Implementation of basic DSP algorithms on digital signal controller Motorola 56F805

Jaroslav Stančík

Bachelor thesis
2008

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky
akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jaroslav STANČÍK**
Studijní program:	**B 3902 Inženýrská informatika**
Studijní obor:	**Automatické řízení a informatika**
Téma práce:	**Implementace základních DSP algoritmů na digitální signální kontrolér Motorola 56F805**

Zásady pro vypracování:

1. **Nastudujte základní výpočetní algoritmy používané při digitálním zpracování signálů se zaměřením na filtraci signálů.**
2. **Seznamte se s hardwarem kitu AP56F801SLK a vývojovým prostředím CodeWarrior IDE.**
3. **Vyberte vhodný typ D/A převodníku a navrhněte způsob jeho připojení k vývojovému kitu.**
4. **Vytvořte knihovnu programových modulů obsahující vybrané algoritmy pro filtraci signálů.**
5. **Ověřte pomocí signálového generátoru a osciloskopu funkci Vámi vytvořených programových modulů.**

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **DAVÍDEK, Vratislav. Implementace algoritmů číslicového zpracování signálů v reálném čase. Praha : ČVUT, 2006. 171 s. ISBN 80-01-03114-4**
2. **Freescale Semiconductor. 56F801 Data Sheet -- Preliminary Technical Data, Rev. 17., 2007. Dostupný z WWW: www.freescale.com**
3. **Freescale Semiconductor. DSP56F801/803/805/807 16-Bit Digital Signal Procesor User's Manual, Rev. 3.0., 2007. Dostupný z WWW: www.freescale.com**
4. **SMÉKAL, Zdeněk, SYSEL, Petr. Signálové procesory. Praha : BEN - technická literatura, 2006. 283 s. ISBN 80-86645-08-8.**
5. **SMITH S. The Scientist and Engineers Guide to Digital Signal Processing, California Technical Publishing, San Diego, California, 1999. Dostupný z WWW: www.dspguide.com**
6. **SOVKA, Pavel. Vybrané metody číslicového zpracování signálů. Praha : ČVUT, 2006. 258 s. ISBN 80-01-02821-6**

Vedoucí bakalářské práce: **Ing. Petr Dostálek**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **22. února 2008**

Termín odevzdání bakalářské práce: **6. června 2008**

Ve Zlíně dne 22. února 2008

prof. Ing. Vladimír Vašek, CSc.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRACT

This thesis is focused on the implementation of the basic low and high pass digital filters on the digital signal controller Motorola 56F805. The synthesis of the digital filters is based on the 1-order IIR filters that simulate the analogue RC filters. The RC filter is described by the transfer function that is converted to the digital IIR filter through applying Z-transform. The filters with the higher order have been created by connecting more IIR 1-order filters in series. The header file of the filter synthesis and IIR filters have been created using the C language. The Matlab Simulink has been also used to create the IIR filters that have been compared with the filters libraries implemented in the DSC 56F805.

Keywords: IIR, DSC Motorola 56F805, DSP, low pass filter, high pass filter, Z-transform, RC filter.

I would like to thank Ing Petr Dostálek for all his direction, assistance, patience and guidance, his recommendation and suggestions that have been invaluable to the project.

In the end, I would like to thank to my family.

Motto

“

It is possible to store the mind with a million facts and still be entirely uneducated.

”

Aleck William Bourne (June 4, 1886 - December 30, 1974)

I hereby declare that I am the sole author of this thesis. The all used literature in this thesis is cited. I will be introduced as the co-author in the case of publication of results, when the licence agreement allows.

In Zlin 29.05.2008 …….

The signature of the bachelor

TABLE OF CONTENT

INTRODUCTION **8**
I . THEORETICAL PART **10**
1 DIGITAL SIGNAL PROCESSING **11**
1.1 DSPS AND ITS BENEFITS **11**
1.2 TYPICAL REAL-TIME DSP SYSTEM **12**
1.2.1 ANALOGUE TO DIGITAL CONVERSION PROCESS 13
1.2.2 SAMPLING THEOREM 14
1.2.3 THE QUANTIZATION ERROR 15
1.3 DSP AND ITS HISTORY **16**
1.4 THE BACKGROUND STUDY **19**
1.4.1 THE MASK 19
1.4.2 TRANSFER FUNCTION 19
1.4.3 Z-TRANSFORM 20
1.4.4 BODE PLOT 20
2 EXPERIMENTAL DSPS SYSTEM **21**
2.1 DIGITAL SIGNAL CONTROLLER MOTOROLA 56F805 ..... **23**
2.2 THE DSP 56F805 CONTROLLER BOARD **25**
3 FILTERS **26**
3.1 RC FILTERS **26**
3.1.1LP RC FILTER 26
3.1.2HP RC FILTER 29
3.2 FIR AND IIR FILTERS **30**
3.2.1 SYNTHESIS OF THE IIR FILTER 31
3.2.2 THE FILTERS WITH THE HIGHER ORDER 33
II. PRACTICAL PART **34**
4 DIGITAL FILTERS **35**
4.1 IIR LP FILTER **35**
4.1.1 SYNTHESIS OF THE IIR 1-ORDER LP FILTER 35
4.1.2 CALCULATION OF THE IIR 1-ORDER LP FILTER 38
4.1.3 IMPLEMENTATION OF THE IIR 1-ORDER LP FILTER 40
4.2 IIR HP FILTER **46**

4.2.1 SYNTHESIS OF THE IIR 1-ORDER HP FILTER....46
4.2.2 CALCULATION OF THE IIR 1-ORDER HP FILTER....48
4.2.3 IMPLEMENTATION OF THE IIR 1-ORDER HP FILTER....49
4.3 IMPLEMEN. OF THE IIR FILTER WITH THE HIGHER ORDER....53
4.3.1 THEORY....53
4.3.2 RESULT AND DISCUSSION....53
5 FILTER LIBRARIES....55
6 THE IMPLEMENTATION OF THE DSC 56F805....58
6.1 FLOW CHART OF THE EXEMPLARY PROGRAM....58
6.2 IMPLEMENTATION OF THE ADC....59
6.3 IMPLEMENTATION OF THE TIMER....61
CONCLUSION....64
TABLE OF USED LITERATURE....65
TABLE OF USED SYMBOLS AND ABBREVIATIONS....68
TABLE OF THE FIGURES....69
LIST OF TABLES....71
TABLE OF APPENDIXES....72

INTRODUCTION

Digital signal processing was born with the availability of digital computers during 1960s and 1970s. Computers were expensive during this era so DSPs was developed for a few applications: radar and sonar, where national security was at risk; oil exploration, where large money could be made; space exploration, where data are matchless; and space exploration, where lives could be saved. During 1980s and 1990s boost and revolution of personal computers started new age of DSPs development for commercial market rather than for military and government needs. It was used widely for public in such products as: mobile telephones, compact disc players, and electronic voice mail. DSPs become graduate course in a graduate level in electrical engineering in the early 1980s. DSPs turn into standard part of the undergraduate curriculum. Today, it is the basic skill needed by engineers and scientist. There are named just few fields influenced by DSPs:

- **Telecommunications:** voice and data compression, echo reduction, signal multiplexing, filtering, video conferencing.
- **Medical:** diagnostic imagining, electrocardiogram analysis, medical image storage/retrieval.
- **Military:** radar, sonar, missile guidance, secure communication.
- **Commercial:** image and sound compression, computer-generated animations in movies, room matching equalization of sound in Hi-fi and sound reinforcement applications, audio effects for use with electric guitar amplifiers, video calling.
- **Industrial:** geophysics, process monitoring and control, non-destructive testing, noise reduction and design tools.
- **Scientific:** earthquake recording and analysis, data acquisition, spectral analysis, simulation and modelling.
- **Space:** space photograph enhancement, data compression.

DSP is a great technology that will have huge impact on science and engineering of twenty-first century [1].

It is obvious how important role the DSPs plays nowadays. The implementation of the basic DSPs algorithm on the DSC 56F805 is the challenge how to deepen the knowledge about the DSPs technology.

In the theoretical part of the thesis can be find information about:

- Digital signal processor and its history,
- Sampling theorem, ADC, quantization error,
- Experimental system used for the real testing,
- Filters as the analogue RC filters, digital IIR and FIR,
- The mask, transfer function, Z-transform and Bode plot.

In the practical part can be found:

- The design, calculation and implementation of the low pass and high pass filters,
- The flow chart of the main program,
- Filter libraries,
- The implementation of the timer and ADC for DSC 56F805.

I. THEORETICAL PART

1 DIGITAL SIGNAL PROCESSING

DSPs can be explained by dictionary terms as follows:

Digital operates with discrete signals that carry information by some numeric value.

Signal is every physical quantity that can be measured in time or even space. It is a variable parameter by which information is conveyed through an electronic circuit.

Processing is the process that performs operations on data according to programmed algorithm.

Put it all together, DSPs is changing or analysing information which is measured as discrete sequences of numbers. The reason why DSPs was invented was computer revolution. It opened new way how to cope with signals in digital world. The information coming from the real word needed to be measured and converted to digital form. Whole process has to be done in the real time [2].

1.1 DSPs and its benefits

Some general advantages of DSPs are:

- Digital system can be *reprogrammed* without any physical removal or changing of electronic components (at least where programmable digital signal processor chips are used),
- System can be easy *reproduce*,
- The system response is not limited by *temperature* and *component tolerances*,
- *Simplicity* [2],

- Guaranteed *accuracy* (it is determined by the number of used bits) ,
- Greater *repeatability*, smaller *size*, lower *cost*, low *power consumption*, and *higher speed* are improving by advanced semiconductor technology.

DSPs has also some disadvantages, nevertheless advanced in new technologies is reducing disadvantages.

- *Speed and cost*. DSPs design can be expensive. Large bandwidth is still processed by analogue method, because the DSP is not fast enough. Also ADC / DAC either are too expensive or do not have adequate resolution. However, DSP devices are becoming faster and faster,
- *Design time.* When the designer is not knowledgeable in DSPs techniques it can be time-consuming to design it.
- *Finite word length* problems. When the not sufficient number of bits is used because of economic consideration. It can result in grim problems of system limitations in real-time situations [3].

1.2 Typical real-time DSP system

Typical simplified real-time DSP system is shown in (fig. 1). It is build from the more blocks. The analogue input filter is situated before the ADC. His function is to reduce aliasing by the bandlimit the analogue input signal. The signal must be hold before entrances the ADC. It is made by a sample and hold circuit. New ADC has included this circuit. The ADC converts analogue signal to digital. The digital signal is filtered in DSP by some mathematical algorithm to produce desired signal that is converted back into analogue signal by DAC. The analogue signal from the DAC can be smoothed by output filter that also removes unwanted high frequencies. In some application input and output filters are not necessary or even ADC or DAC. For example, when the processed data from DSP are saved in the computer [3].

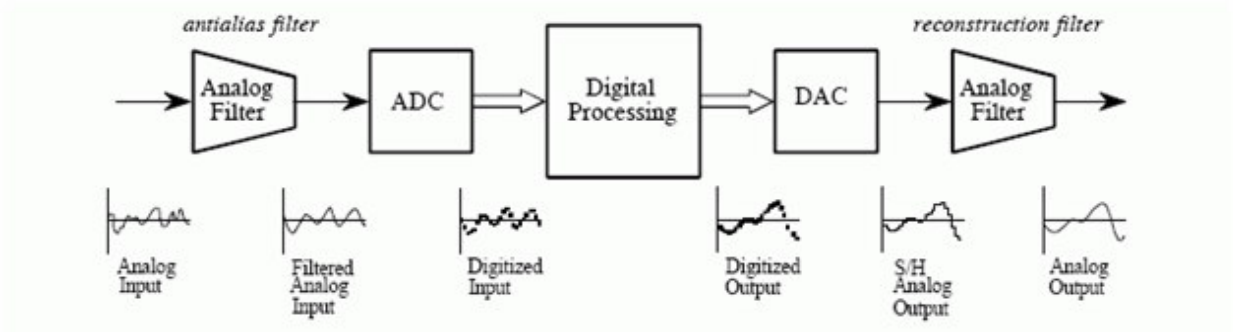


***Figure 1**: The diagram of general real-time DSP system [2].*

1.2.1 Analogue to digital conversion process

The analogue signal has to be converted to digital signal because the most application of DSP can not deal with the analogue signal. The conversion has the following steps (fig. 2):

- The *analogue input* signal (it can be bandlimited from filter) is first sampled. It changes analogue signal into discrete-time continues amplitude signal (*sampled signal*).
- Each single sample of amplitude of discrete-time continues signal is quantized into one of 2^B levels. The B is bits depth of ADC [3]. It is usually represented as 8-bit (256 levels), 16-bit (65,536 levels), 32-bit (4.3 billion levels), and so on, though any number of quantization levels is possible, not just powers of two [3].
- The discrete-amplitude levels are encoded into binary world with length B bits (*digitalized signal*).

This can give us three types of signals:

- *Analogue* signal that is continues in both time and amplitude. It has infinite precision [4].
- *Sampled* signal is continues in amplitude, but identify just discrete values in time

- The *digital* signal, x (n) (n=0, 1, …). The signal exists just in discrete points in time where it has some numeric value that is generally some multiple of some basic constant value (2^B) [1]. It can be also called discrete-time discrete-value signal [3]. This type of the signal is concern of this thesis.

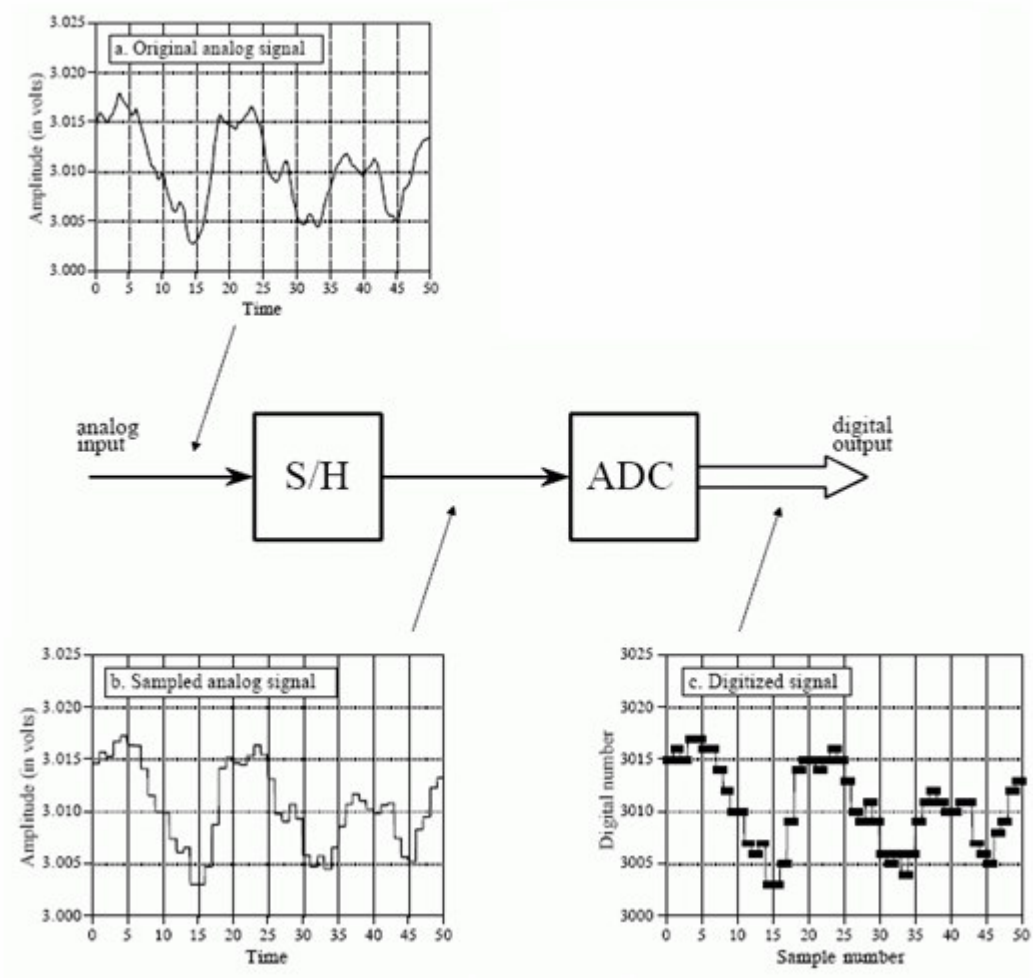


Figure 2: *The sampling of the analogue signal is showed. Foremost the signal is sampled. After that is hold because of keeping voltage at the constant value while conversion is taking place in ADC. Then it is quantized by ADC [2].*

1.2.2 Sampling theorem

The proper sampling is when is possible to reconstruct original analogue signal from sampled signal without any lose of information.

The general rules how to avoid any lose of signal is characterized by the Shannon-Kotelnikov-Nyquist sampling theorem. It stated that original signal must be sampled at the rate of at least two-times higher frequency as is his highest frequency f_{MAX} ($f_s \geq 2\ f_{MAX}$ where f_s is the sampling frequency).

The proper sampling is showed in the first picture, where the sampling frequency is at least two times higher than the frequency of the original signal. The improper sampling is showed in the second picture, where the sampling frequency is lower than sampling theorem that leads to the loss of the original signal. It creates new signal with totally different frequency from the original one (fig. 3). It is called an aliasing [2].

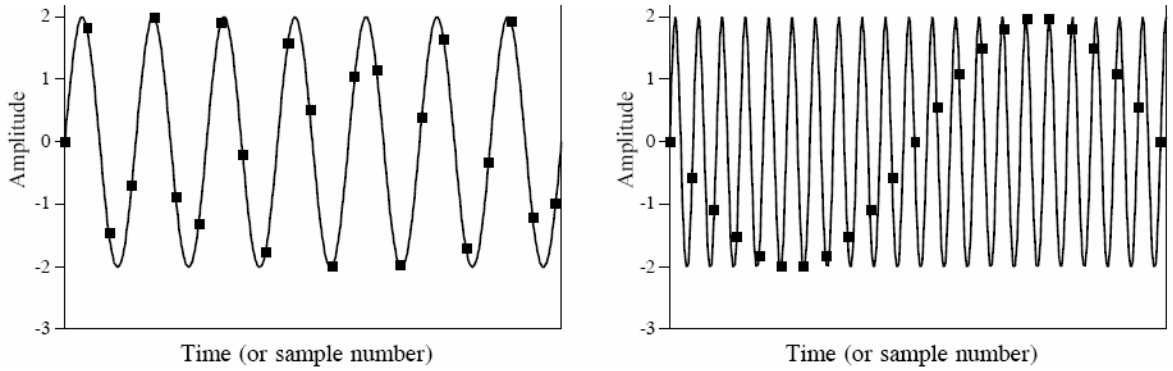


Figure 3: *In the first picture is showed the proper sampling. The aliasing effect caused by the improper sampling frequency is showed in the second picture [2].*

1.2.3 The quantization error

The quantization error originates when ADC is converting voltage to the nearest integer [2]. The quantization errors also occur during conversion due to the inaccuracy in the measurement, uncertainty in the timing, and limit on the duration of measurement [4]. Quantization noise is the difference between the blue and red signals in the upper graph. The

noise is called the quantization error, which is "added" to the original signal and is the source of the noise (fig. 4).

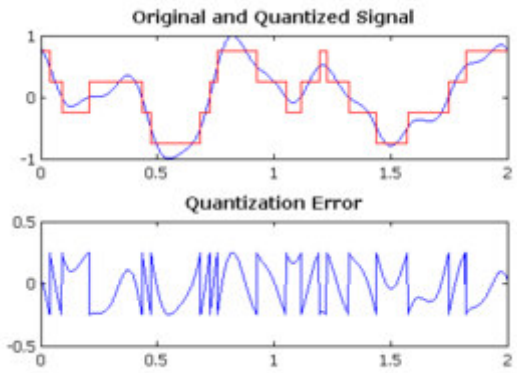


Figure 4: *Quantization error [5].*

1.3 DSP and its history

A DSP is a specialized microprocessor designed specifically for the digital signal processing, generally in real-time computing [6]. Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly on a set of data. DSP has to gone through some development to achieve the required processor performance. Among the most important thing that are characteristic for the DSP are: architecture, program flow, memory structure, data operation, instruction set.

At first DSP was constructed from parts of DSP and some special components (crystal, sampled circuit capacitor etc. The company INTEL first made signal processor i2920/21 in 1979. It was attempt to made DSC from adapting the structure of single chip microcontroller to signal processing. The DSP did not find wider usage because of untouched sequence of analogue operations. Inner numeric processing of signal was just made in one or some blocs. 9 bits A/D and D/A was placed on the chip. The analogue and numeric circuit had to be separated with care owing to the noise of numeric circuit that could change the precision of A/D conversion. The chip missed hardware multiplier and quarter-squares multiplier could be just realized by programming. For these all

disadvantages, the i2920/21 has simply programmable code that allow realize more complex algorithm. For example, numeric IIR filter up to 20 orders for sampling frequency around 13 KHz.

Really first DSP with hardware architecture was µPD 7720 made by company NEC in 1980. It assured the requirements of processing numeric signals in real time. From this time, the signal processors can be generally divided into three groups [1].

DSP of first generation (1979-1985) are featured by hardware architecture, and so had separate instruction and data memory [6]. It contains hardware multiplier. ALU makes calculations just in fixed-point (integer) arithmetic. It already had a special instruction set, with instructions like load-and-accumulate or multiply-and-accumulate. The higher operation speed is achieved by using more inner buses. It is not possible to connect external memory to internal memory (exception was TMS 32010). The programmable code was extremely complicated. The communication with outer circuit is realized by serial or parallel canal. The processors were made by the NMOS technology. The DSP works in infinite loop which is stopped by incoming input sample. This shows that DSP makes just calculation of process signal algorithm and others operations have to be made by higher processors or computer. Among representatives of this generation can be named these types:

- i2920 (INTEL),
- µPD 7720,
- TMS322010 (TEXAS INSTRUMENTS).

DSP of second generation (1983-till now) has also arithmetic in fixed-point arithmetic with word length 16 and 24 bits. But it was rapidly decreased length of instruction cycle (down to one third and less). It has pipelining architecture of processing data. Even, some types had doubled or troubled hardware architecture (family DSP56K). Memory for coefficient (type ROM) and memory for intermediate data (type RAM) were united to the one block. Memory data blocks include individual address units that can generate addresses that are

needful for realization of the fast Fourier transform. Contact circuits can work independently. They can enclose A/D and D/A convectors. This device can cooperate with common types of 16 or 32 microprocessors. It is possible to connect more external memories on one signal processor that enables to realize different algorithm of processing. Energy composition was put down by CMOS technology. C language can be used in modern types of DSP. The producers supply circuit emulator on the chip that makes easier to tune algorithm. Among most used types belongs:

- Family DSP56K (MOTOROLA),
- TMS320C25, family TMS320C50 etc. (TEXAS INSTRUMENTS),
- Family ADSP2100 (ANALOG DEVICES),
- DSP16A (AT &T),
- µPD 77220 (NEC) etc.

DSP of third generation (1987-till now) works with floating-point arithmetic with 32 bits operands. They have a lot of properties of second generations DSP. The difference from second generation is in higher dynamic and higher precision of signal. On the other hand they are more expensive. Instruction set includes instructions for as for fixed-point architecture so for floating-point architecture [1]. Among them belong:

- DSP96002 (MOTOROLA),
- TMS320C30 (TEXAS INSTRUMENTS),
- ADSP21000 (ANALOG DEVICES),
- DSP32C (AT&T),
- µPD 77220 (NEC) etc.

1.4 The Background study

1.4.1 The mask

In the computing the mask is used for bitwise operations to set either on or off. In other words, the purpose of the mask is set a single bit in the register to logic 0 or logic 1. The logic function OR is used to set bit to 1 and the function AND is used to set bit to 0 (tab. 1). The mask can be written in different numeric format. Table shows bitwise operations written in binary and hex format.

Binary mode																Name	Ope.	Hex mode
0	1	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	1	**Reg.**	**AND**	5005
1	1	1	**1**	1	1	1	1	1	1	1	1	1	**0**	1	1	**Mask**		fffb
0	1	0	**1**	0	0	0	0	0	0	0	0	0	**0**	0	1	**Reg.**		5001
0	1	0	**1**	0	0	0	0	0	0	0	0	0	1	0	1	**Reg.**	**OR**	5005
1	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	**Mask**		8000
1	0	0	**1**	0	0	0	0	0	0	0	0	0	1	0	1	**Reg.**		9005

Table 1: *The table show the usage of the bitwise operations. The upper table shows the logical function AND that is usually used for setting the bit to logical 0. On the other hand, the logical function OR is used for setting the bit to logical 1. The masks are written in the binary and hex formats.*

1.4.2 Transfer function

A transfer function is a mathematical representation of the relation between the input and output amplitudes of a linear time-invariant system. The transfer function fully reveals how the circuit modifies the input amplitude in creating the output amplitude. The transfer function allows the completely mathematical simulation of the real system. It is commonly used for the mapping the single-input single-output systems as for example electronic RC filters are [7].

1.4.3 Z-transform

The applications of the z-transform are many. In this thesis, it is focused on designing recursive digital filters that simulate the performance of the analogue filters [3]. The z-transform is the mathematical way how to compute the filter coefficients from the transfer function. This technique solution of the z-transform is derived in practical part of the thesis.

1.4.4 Bode plot

In this thesis, it is used to show the transfer function or frequency response of a linear, time-invariant system. It helps to characterize the properties of the filters. It is usually a combination of a Bode magnitude plot and Bode phase plot (fig. 5). A Bode magnitude plot is a graph of log magnitude versus frequency, plotted with a log-frequency axis, to show the transfer function or frequency response of a linear, time-invariant system. The magnitude is usually expressed in the decibels. A Bode phase plot is a graph of phase versus frequency, also plotted on a log-frequency axis [8].

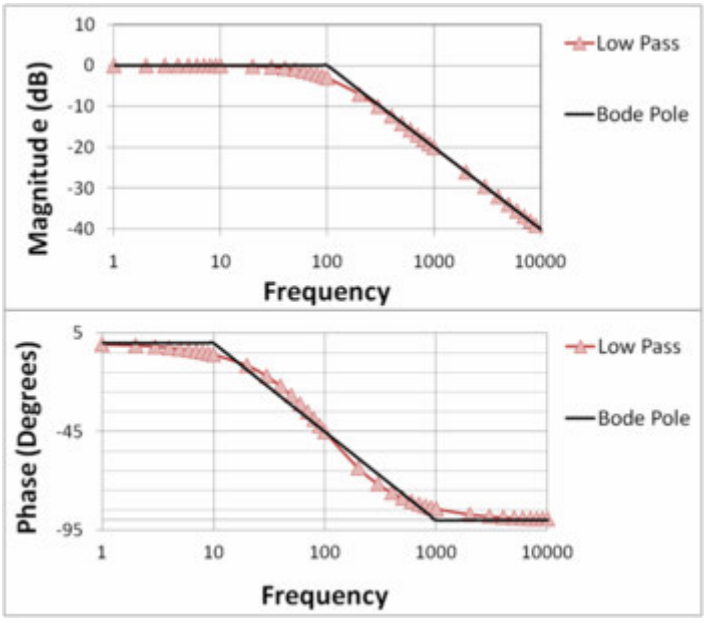


Figure 5: *The properties of the LP filter are characterized by the Bode diagram [8].*

2 EXPERIMENTAL DSPS SYSTEM

The experimental system DSPs was designed and constructed by the supervisor of this thesis. The system consists of these components:

- The digital signal controller Motorola 56F805,
- The DSP56F805 Controller Board,
- The oscilloscope Gw Instek GDS-820S (fig. 6),
- Waveform Generator ETC-Arbitrary (fig. 7),
- The printed circuit DAC board (it was made by the supervisor of the thesis),
- Two personal computers.

- **The oscilloscope**

Manufacturer: Instek

Model: GDS-820S

More information can be found [9].

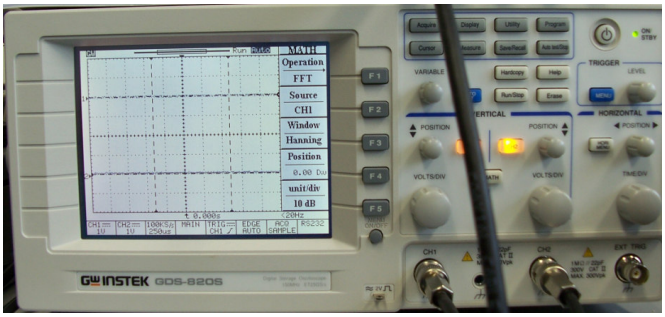


Figure 6: *The basic parameters of the oscilloscope.*

- **Waveform Generator**

Manufacturer: ETC s.r.o.

Model: ETC M631

More information can be found [10].

Figure 7: *The basic parameters of the waveform generator.*

The DSPs system map is explained below (fig. 8):

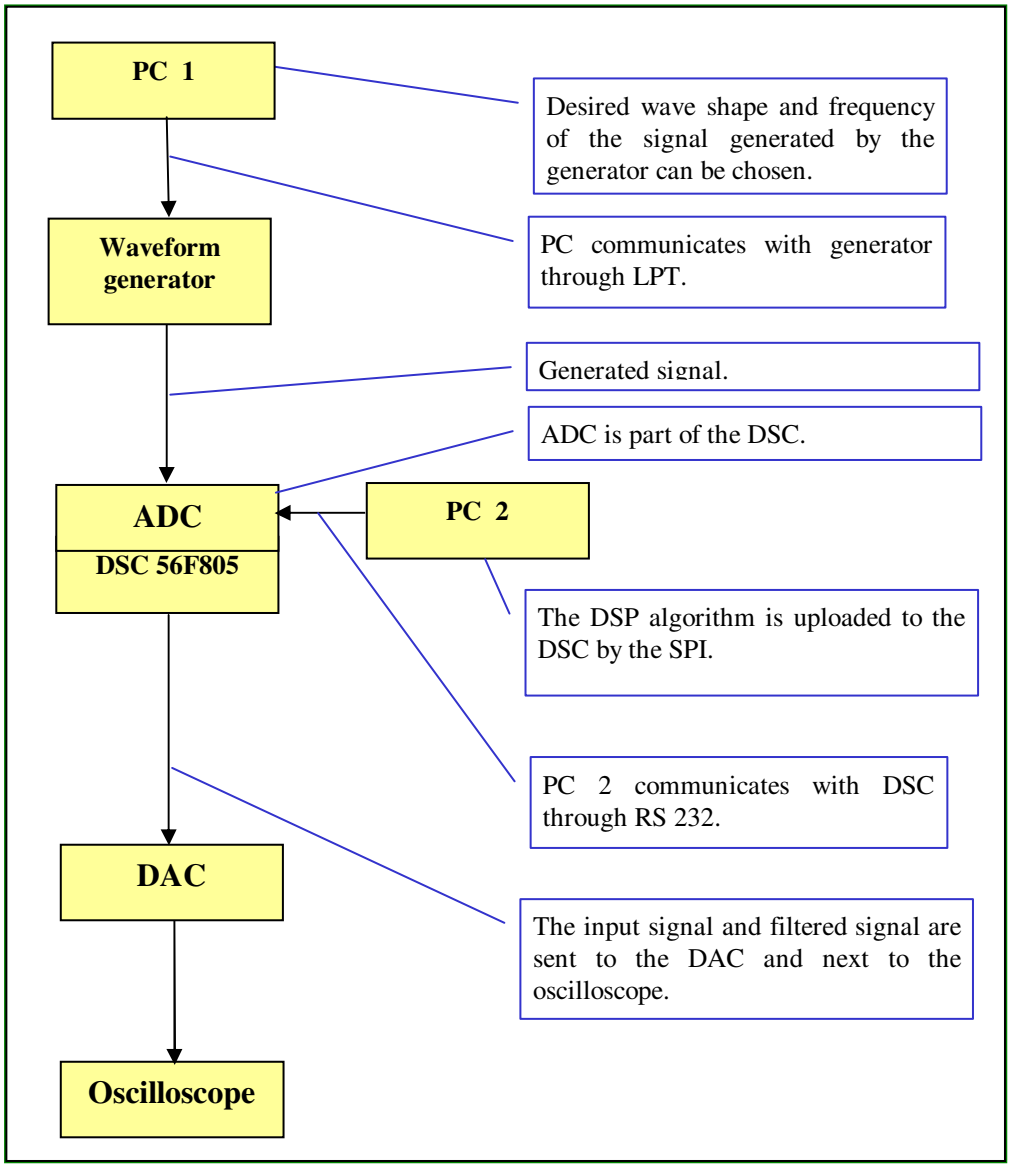


Figure 8: *The system map of the experimental DSPs system.*

The first PC manages the setting of waveform generator by the generator software (fig. 9). The communication is through the LPT. The generator generates desired analog waveform that is sent to the DCP. Before that the analog signal is converted to the discrete signal by the ADC that is part of the DSP 56F805 Controller board. The digital filtration is made in the DSC core by the desired DSPs algorithm. The filtered signal processed by the core is converted back to the analogue signal by the DAC that is part of the printed circuit board.

The both original and filtered signals are compared in the oscilloscope. The communication between second PC and DSC is across RS 232.

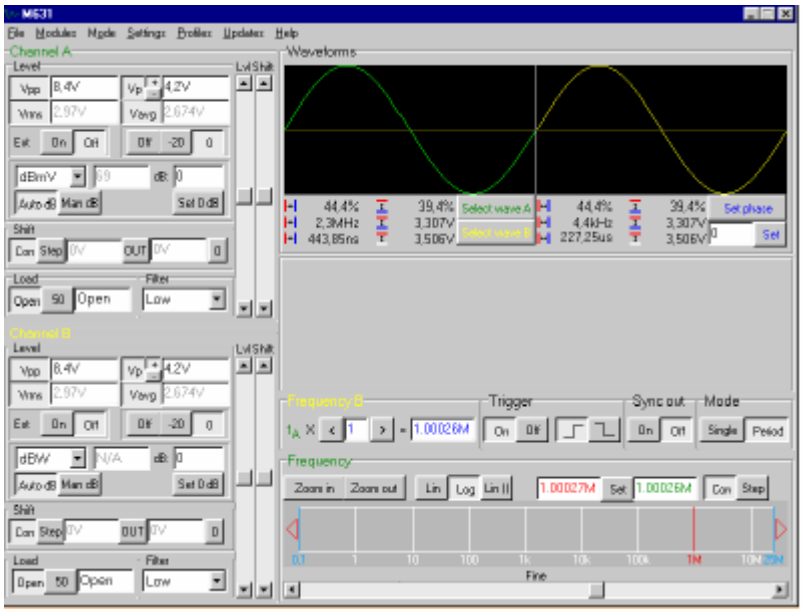


Figure 9:*The generator software that serves for example, for the setting the desired shape of the waveform, the frequency etc.*

2.1 Digital signal controller Motorola 56F805

56F805 is based on 56800 core-based family processors. It is combination of DSP processor and microcontroller. That made it extremely cost-effective solution for many applications. The core of 56800 is based on hardware-style architecture. The microprocessor-style programming model and optimized instruction set allow generation of efficient code for both microcontroller and DSPs applications. The instruction set is also very proficient for C compilers. The 56F805 controller includes these types of memories (fig. 10):

- Program flash (31.5K words) and data flash (4K words) (each programmable through the JTAG port) – Data, program and booth flash can be independently bulk-erased or erased in page sizes of 256 words,
- Program RAM (512 words),
- Data RAM (2K words),
- External Memory (64K) - it supports program execution,
- Boot Flash (2K words) – can be used to program the main program and data flash memories.

This device contains a full set of standard programmable peripherals (fig. 10):

- Two Serial Communications Interfaces (SCI),
- One Serial Peripheral Interface (SPI),
- ADC,
- Four Quad Timers (any of these interfaces can be used as General Purpose Input/Outputs (GPIOs) if that function is not required),
- Controller Area Network interface (CAN Version 2.0 A/B-compliant),
- Internal interrupt controller [11].

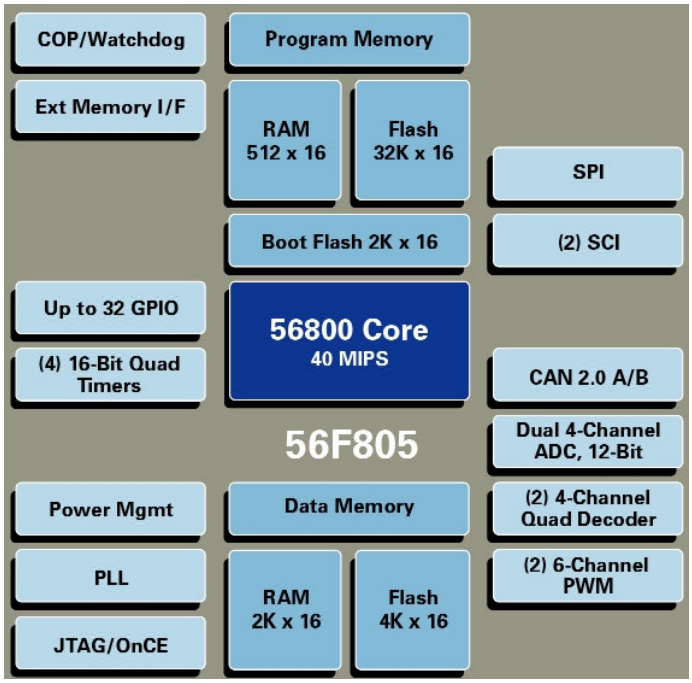


Figure 10: *The Block Diagram of the DSC 56F805 [11].*

2.2 The DSP 56F805 Controller board

The DSP56F805 Controller Board is used to demonstrate the abilities of the DSP56F805. It is an evaluation module type of board that includes a DSP56F805 part, resolver interface, digital and analog power supply including precise reference for the on-chip A/D converter and peripheral expansion connectors. It is designed to allow new users to become familiar with the features of the 56800 architecture [12].

3 FILTERS

It is a device that changes the shape of the wave, amplitude-frequency and/or phase-frequency characteristic of the signal. The filters can be realized electrically by the analogue devices as for example simple RC circuit or digitally by some mathematical algorithm implemented in the hardware.

Basically, the analogue filters filter the analogue signal based on the way how the voltage changes across the RC circuit. The voltage changes are dependent on the circuit properties. The precision of the analogue filters is limited by the characteristic of the used components.

Digital filters operate with the digitized signal that is stored in the computer memory. The type of digital filtration depends on the used algorithm. The precision of the digital filters is limited by the word length used [3].

3.1 RC filters

This chapter contains the description of the analogue RC filters, because they are used to design the digital filters in this thesis.

3.1.1 LP RC filter

Low pass filter is the filter that let frequencies below corner frequency to pass and stop frequencies above corner frequency. The most basic LP filter can be created by the single resistor-capacitor circuit (fig. 11).

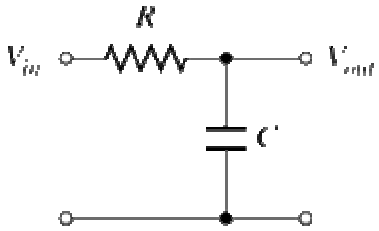


Figure 11: *The LP RC filter.*

The properties of the filter can be explained in the time-domain. The voltage on the R decreases towards zero as time passes. On the other hand, voltage on the C is charged towards voltage V (fig. 12). The circuit becomes opened when the capacitor is fully charged with the voltage V. The time constant is time when the circuit becomes opened. It can be calculated by the equation 1 [16]. The corner frequency can be calculated by the equation 2.

$$T = R \cdot C \ [s] \tag{1}$$

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C} [Hz] \tag{2}$$

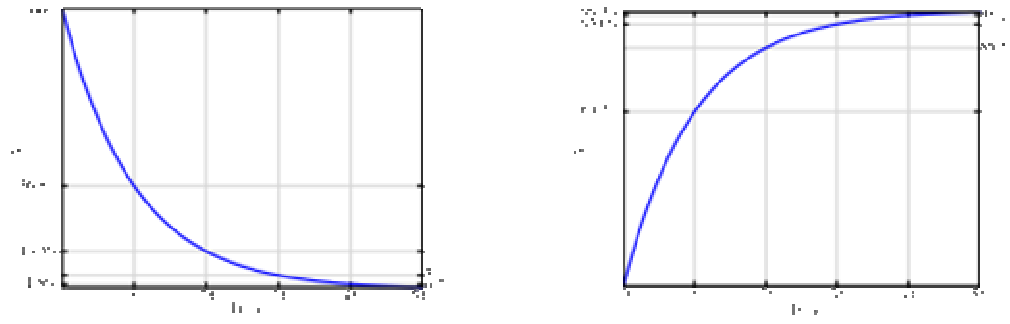


Figure 12: *In the first picture is showed discharging of the resistor with time passing. On the other side the capacitor is charging as time goes. The circuit becomes opened when the capacitor is charged. The time constant is dependent on the resistor and capacitor value.*

The properties of the filter can be also explained in frequency-domain by the so called Bode plot (fig. 13). The reason to do this is to show how the filter behaves for different angular frequencies. General interest is put on the magnitude of the response.

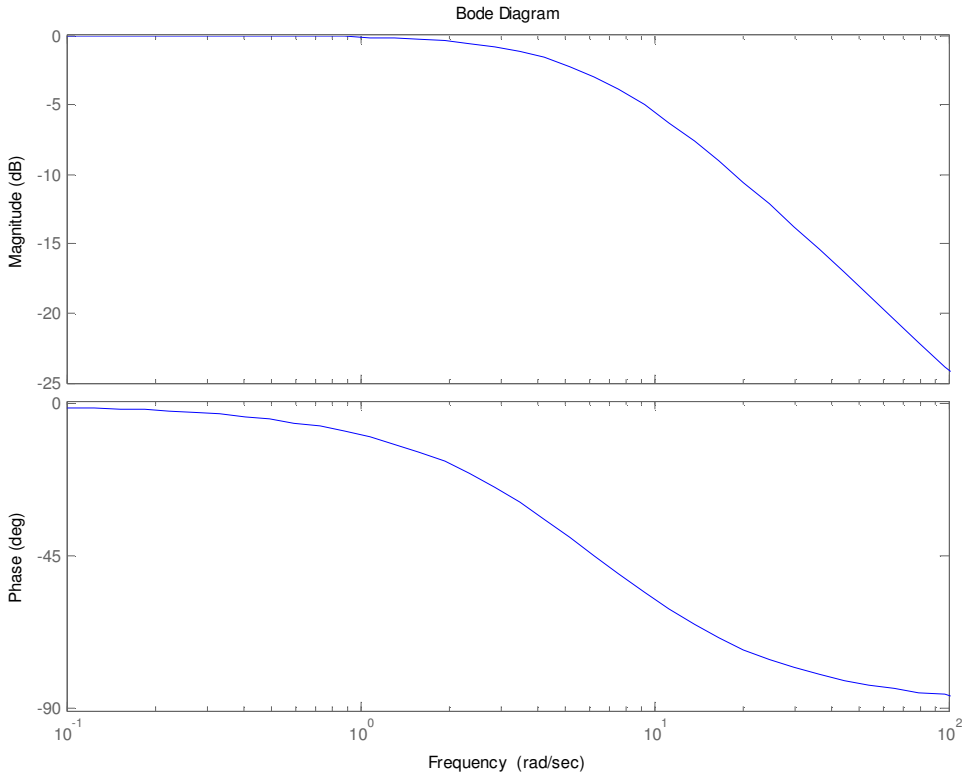


Figure 13: *The gain and the phase of a system are showed as a function of frequency. In the top graph is showed the decay of the signal magnitude that is caused by the 1-order LP filter. The decay starts when the input signal frequency is higher then cut off frequency of the LP filter. In the bottom graph is showed the phase shift of the signal.*

The input angular frequency (ω) can be calculated by the equation 3 [rad/s]. As was previously mentioned $T = R \cdot C \ [s]$, T can be also calculated by $T = \frac{1}{\omega} \ [s]$ which is $\omega = \frac{1}{T} [\text{rad/s}]$, so:

$$\omega = \frac{1}{R \cdot C}[s] \tag{3}$$

For ω < 1/RC the output voltage is equal to input, so the signal is passed unchanged. The output voltage is attenuated when the input voltage is equal to ω = 1/RC. It is called corner frequency. When the ω > 1/RC the capacitor acts as a short circuit and all voltage is across the resistor [17].

3.1.2 HP RC filter

It is the filter that let frequencies above corner frequency to pass and stop frequencies below corner frequency. The most basic first 1-order high pass filter can be created by the single RC circuit [16]. The scheme of the filter is almost same like scheme of the LP filter, but resistor and capacitor are changed instead of each other (fig. 14). The corner frequencies and time constant of the filter are dependent on the chosen values of resistor and capacitor. They can be calculated by the equations 1 and 2 from the previous chapter.

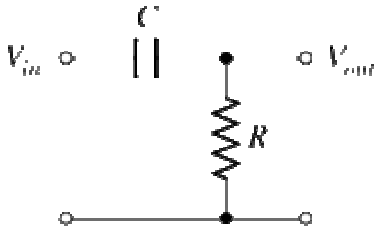


Figure 14: *The HP RC filter.*

The properties of the filter described in the frequency domain:

At high frequencies ω > 1/RC, the capacitor acts as a short circuit and the gain is 1 (the signal is passed). At low frequencies, ω < 1/RC, the capacitor is an open and the output is zero (fig. 15) [17].

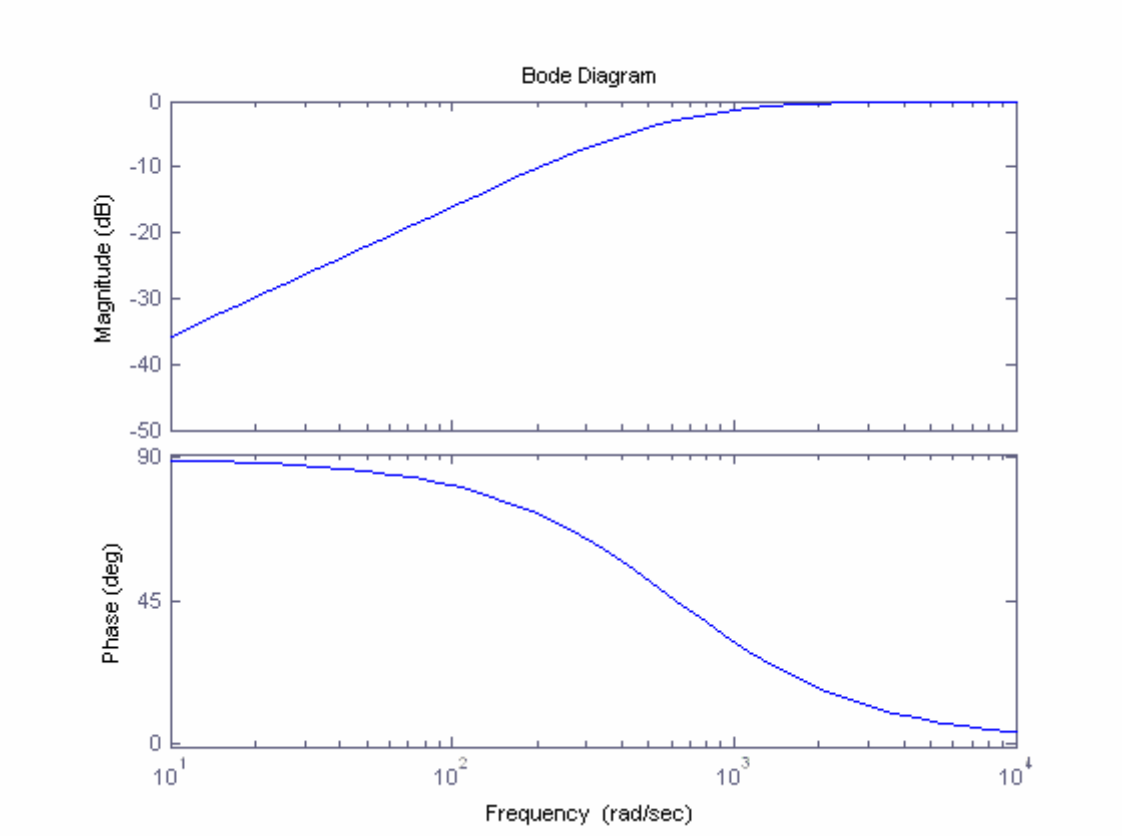


Figure 15: *The gain and the phase of a system are showed as a function of frequency. In the top graph is showed the ascent of the signal magnitude that is caused by the first order HP filter. The ascent starts when the input signal frequency is higher then cut off frequency of the filter. In the bottom graph is showed the phase shift of the signal.*

3.2 FIR and IIR filters

The digital filters can be divided into two main groups IIR and FIR filters (tab. 2).

FIR filters are digital filters that express each output sample as a weighted sum of the last N inputs, where N is the order of the filter. They are stable because they do not use the feedback as IIR filters do. The main disadvantage is that they may require significantly more processing and memory resources than cleverly designed IIR variants. In theory the impulse

response of the filter has finite length, hence the name FIR. The FIR can not simulate the analogue filters as IIR can. The simplest FIR filter is moving average [13].

IIR filters are digital counterpart to the analogue filters. The filter output is dependent on the previous input and previous filtered output. In other words, they use feedback. In theory the impulse response never dies out completely, hence the name IIR. From the experimental data is proved, that for the same task the order of the IIR filter is 10 times smaller then FIR order [14]. So the IIR normally needs less computing then FIR filters with the same properties. However, due to the feedback, high order IIR filters may have problems with instability, and arithmetic overflow, so require careful design to avoid such pitfalls [13].

FIR		**IIR**	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
Still stable.	High order.	Low order.	Problems with stability.
The chance of the arithmetic saturation is lower.	High time delay during data processing.	Low time delay during data processing.	The arithmetic saturation of processor can occur, because of the feedback
Appropriate for the adaptive algorithm.	High data memory requirements.	Low data memory requirements.	It is hard to use for the adaptive algorithm.

Table 2: *The table shows the basic differences between FIR and IIR filters [14].*

3.2.1 Synthesis of the IIR filter

The method how to design digital filter is to simulate the simple RC analogue filter. Basically, the analogue filter is converted to the digital by applying some discretization technique. First, the analogue RC filter is characterized by the transfer function. The transfer function fully simulates the behaviour of analogue filter. The image of the transfer function is gained by the Z-transform. The recursive equation for the digital filter is derived from the image of the transfer function. This technique is more explained in the practical part of the

thesis. By this technique is possible to design just IIR filter. The design of the FIR filter is out of the scope of this thesis. The IIR 1-order LP filter derived by the Z-transform is represented by the universal recursive equation $Y(kT) = a_0 \cdot Y(kT-1) + b_0 \cdot U(kT-1)$, where u is input signal, y is output signal, and k are singular steps of the filtration. The filtered samples of the signal are dependant on the value of coefficients a_0 and b_0. The recursive equation for the IIR 1-order HP filter derived by the Z-transform looks as $Y(kT) = a_0 \cdot Y \cdot (kT-1) + U - b_0 \cdot U \cdot (kT-1)$. The performance of the digital filter is the same as the performance of the analogue RC filter (fig. 16). The properties of the filer are dependent on the coefficients a_0 and b_0 .

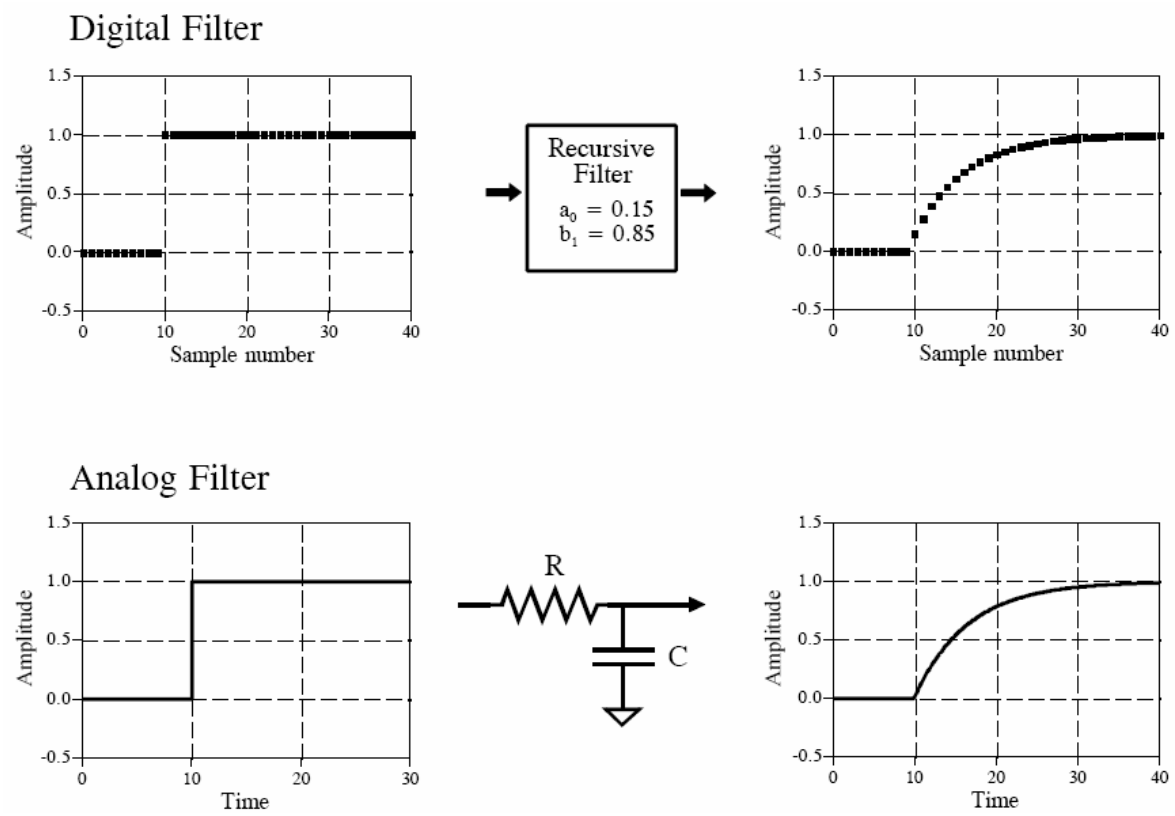


Figure 16: *Digital 1-order LP recursive filter smoothes the edge of a step input, just as an electronic RC filter [2].*

3.2.2 The filters with the higher order

The higher order filters have much better attenuation properties then lower-order filters. The higher order filter can be created by the connecting more 1-order filters in series (fig. 17). The advantage of this method is really simple realization. The highest order of the filter that can be achieved by this technique is limited by the fall of the voltage on the output caused by the resistors, capacitors and cables.

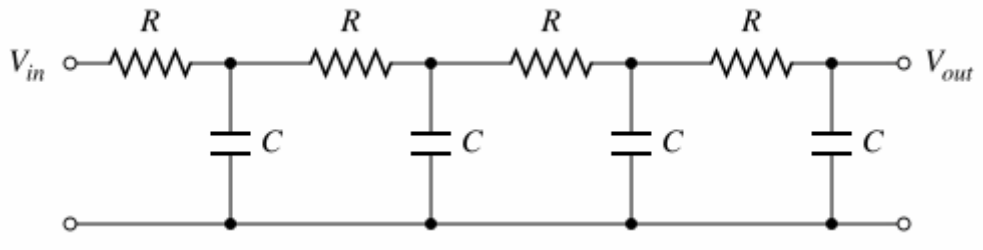


Figure 17: *The 4-order analogue LP filter is created by the four 1-order analogue LP filters connected in series one by one.*

The digital higher-order filter can be created by the same method as the analogue one (fig. 18). The highest order of the filter that can be reached by this technique is limited by the rounding errors made during calculations, quantization errors etc.

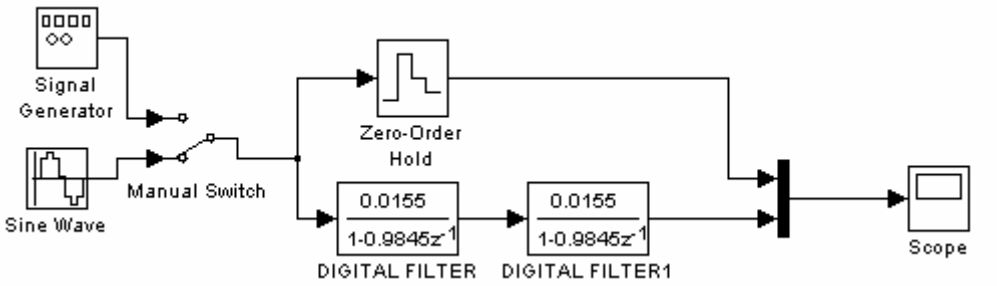


Figure 18: *The 2-order IIR LP filter is created by the two 1-order IIR LP filters connected in series one by one. The coefficients of the filter are* $b_0 = 0.0155$ *and* $a_0 = -0.9845$. *The scheme is created in the Matlab Simulink.*

II. PRACTICAL PART

4 DIGITAL FILTERS

4.1 IIR LP filter

4.1.1 Synthesis of the IIR 1-order LP filter

To design the filter one has to do the following steps:

1. *The cut off frequency:*

First, the desired corner frequency $f_c[Hz]$ is chosen.

2. *The angular frequency:*

Second, the angular frequency is calculated (equation 4).

$$\omega = 2 \cdot \pi \cdot f_c[rad/s] \tag{4}$$

3. *The time constant:*

After that the time constant is derived from the equation 5. The time constant is exactly same as time constant for the analogue RC filter which is $T = R \cdot C[s]$ (equation 1).

$$T = \frac{1}{\omega}[s] \tag{5}$$

4. *The transfer function:*

The transfer function of the analog low pass RC filter is determined by the ratio of the voltage output and input in the equation 6 (the derivation of this equation is out of scope of this thesis). The transfer function of the low pass filter is also defined in the equation 7.

$$\frac{V_{OUTPUT}}{V_{INPUT}} = \frac{\frac{1}{s \cdot C}}{\frac{1}{s \cdot C} + R} = \frac{1}{1 + s \cdot C \cdot R} \tag{6}$$

$$G(s) = \frac{1}{1 + T \cdot s} = \frac{\frac{1}{T}}{\frac{1}{T} + \frac{T \cdot s}{T}} = \frac{\frac{1}{T}}{\frac{1}{T} + s} = \frac{Y(s)}{U(s)} \tag{7}$$

5. *The Z-transform:*

The Z-transform with the zero-order holder (equation 8) is used to gain the image of the transfer function in the complex plane, with sampling frequency (f_s) at least two times higher as the highest used frequency (see sampling theorem). The image of the transfer function will contain the filter coefficients a_0 and b_0.

$$G(z,\varepsilon) = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{G(s)}{s} \right\} = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{s} \cdot \frac{\frac{1}{T}}{\frac{1}{T} + s} \right\} \tag{8}$$

Partial fraction decomposition is used to calculate coefficients A and B (equation 9).

$$\frac{1}{s} \cdot \frac{\frac{1}{T}}{\frac{1}{T} + s} = \frac{A}{s} + \frac{B}{\frac{1}{T} + s} \tag{9}$$

$$A \cdot \left(\frac{1}{T} + s \right) + B \cdot s = \frac{1}{T}$$

$$s = -\frac{1}{T} : -\frac{1}{T} \cdot B = \frac{1}{T} \Rightarrow B = -1$$

$$s = 0 : \frac{A}{T} = \frac{1}{T} \Rightarrow A = 1$$

The Z - transform dictionary is used in the transform. Relative shift is chosen $\varepsilon = 0$ (equation 10).

$$G(z,\varepsilon) = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{s} \cdot \frac{\frac{1}{T}}{\frac{1}{T}+s} \right\} = \frac{z-1}{z} \cdot \left\{ \frac{z}{z-1} \cdot \frac{z \cdot e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\cdot\varepsilon\right)}}{z - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}} \right\} = 1 - \frac{z-1}{z - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}} = \tag{10}$$

$$= \frac{z - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)} - z + 1}{z - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}} = \frac{1 - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}}{z - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}} = \frac{b_0}{a_1 \cdot z - a_0}$$

6. *The filter coefficients:*

The algorithm for the calculation of filter coefficients is derived from the equation 10. It is showed in the equation 11.

$$b_0 = 1 - e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)} \tag{11}$$

$$a_0 = -e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}$$

$$a_1 = 1$$

7. *The 1 order LP IIR filter:*

The recursive equation for the first order LP IIR filter (equation 12) is gained by the z-transform (equation 10), where a_0 and b_0 are coefficients of the filter, u is input signal, y is output signal, and k are singular steps of the filtration.

Negative power of z is applied:

$$\frac{b_0}{a_1 \cdot z - a_0} = \frac{Y(z)}{U(z)} \tag{12}$$

$$\frac{b_0 \cdot z^{-1}}{a_1 \cdot z \cdot z^{-1} - a_0 \cdot z^{-1}} = \frac{Y(z)}{U(z)}$$

$$Y(kT) = a_0 \cdot Y \cdot (k \cdot T - 1) + b_0 \cdot U \cdot (k \cdot T - 1)$$

4.1.2 Calculation of the IIR 1-order LP filter

Task:

The LP filter can be used in the stereo equalizer when just low frequencies (basses) are passed and high frequencies (trebles) are banned. Everything that is higher than 100 Hz (f_c) will be attenuated. The LP filter can be designed as followed:

Solution:

1. *The cut off frequency:*

$f_c = 100Hz$

2. *The angular frequency:*

$\omega = 2 \cdot \pi \cdot f_c = 2 \cdot \pi \cdot 100 = 628 rad / s$

3. *The time constant:*

$T = \frac{1}{\omega} = \frac{1}{628} = 0.0016$

4. *The transfer function:*

$G(s) = \frac{1}{1 + 0.0016 \cdot s} = \frac{Y(s)}{U(s)}$

5. *The Z-transform:*

The Z-transform is calculated below (ε = 0, zero-order holder) with the sampling frequency two times higher as 20 kHz, so $f_s = 40kHz$ (common range of the human hearing is from 20 Hz to 20 KHz).

$$G(z,\varepsilon) = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{s} \cdot \frac{1}{1+0.0016 \cdot s} \right\} = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{\frac{1}{0.0016}}{s \cdot \left(\frac{1}{0.0016} + \frac{0.0016 \cdot s}{0.0016} \right)} \right\} =$$

$$= \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{628}{s \cdot (628+s)} \right\}$$

Partial fraction decomposition:

$$\frac{628}{s \cdot (628+s)} = \frac{A}{s} + \frac{B}{(628+s)}$$

$$A \cdot (628+s) + B \cdot s = 628$$

$$s = -628 : -628 \cdot B = 628 \Rightarrow B = -1$$

$$s = 0 : 628 \cdot A = 628 \Rightarrow A = 1$$

$$G(z,\varepsilon) = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{s} \cdot \frac{\frac{1}{T}}{1+628s} \right\} = \frac{z-1}{z} \cdot \left\{ \frac{z}{z-1} \cdot \frac{z \cdot e^{-628 \cdot \frac{1}{40000} \cdot 0}}{z - e^{-628 \cdot \frac{1}{40000} \cdot 0}} \right\} = 1 - \frac{z-1}{z-0.9845} =$$

$$= \frac{z-0.9845-z+1}{z-0.9845} = \frac{0.0155}{z-0.9845}$$

6. *The filter coefficients calculated by the algorithm are:*

$$b_0 = 1 - 0.9845 = 0.0155$$
$$a_0 = -0.9845$$

The control of the Z-transform calculation is done in the Matlab by the command c2dm. The c2dm has some parameters as method of conversion from LTI to discrete time ('zoh' is

chosen in this example) and period (1/40 000). The more about this command can be find in Matlab Help.

Whole command looks like:

num = 1
den = [0.0016 1]
[num1,den1] = C2DM(num,den,1/40000,'zoh')

The coefficients calculated by Matlab are also $a_0 = -0.9845$ and $b_0 = 0.0155$.

7. *The 1 order LP IIR filter (equation 13):*

Negative power of z is applied:

$$\frac{0.0155}{z - 0.9845} = \frac{Y(z)}{U(z)}$$

$$\frac{0.0155 \cdot z^{-1}}{z \cdot z^{-1} - 0.9845 \cdot z^{-1}} = \frac{Y(z)}{U(z)}$$

$$\frac{0.0155 \cdot z^{-1}}{1 - 0.9845 \cdot z^{-1}} = \frac{Y(z)}{U(z)}$$

$$Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + 0.0155 \cdot U \cdot (k \cdot T - 1) \quad (13)$$

4.1.3 Implementation of the IIR 1-order LP filter

4.1.3.1 Theory

The designed and calculated IIR 1-order LP filter from the previous chapters is coded to universal digital filter by the C language. It is uploaded to the DSC core by the SPI. The

input signal from the waveform generator and filtered signals from DSC are compared in the oscilloscope. The tested input signal has different shapes of the waveform as sine, square and sawtooth (fig. 19).

First, the filter is tested for the input signal with frequency 40 Hz. This signal shall be passed with the minimal amplitude loss. The phase shall not be changed when the signal is passed through the filter.

The second, the 1 k Hz signal is send through the filter. The filtered signal shall be attenuated at and his phase shall be changed. These filtrations are compared with the same filtration process done in the Matlab Simulink.

The recursion equation of the tested filter is in the equation 13. It is IIR 1-order LP filter designed for the $f_c = 100Hz$, and $f_s = 40kHz$.

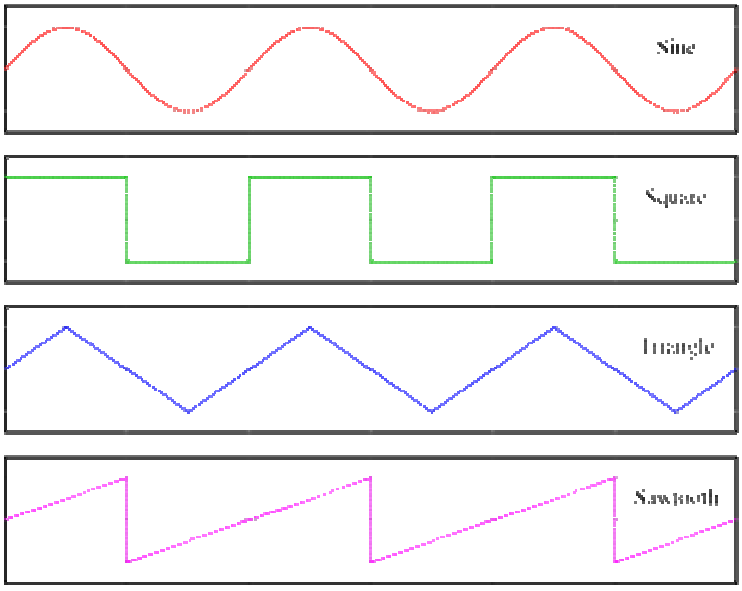


Figure 19: *The different shapes of the waveforms are showed in the figure [15].*

The digital filter is also tested in the Matlab Simulink to known if the filter operates correctly (fig. 20).

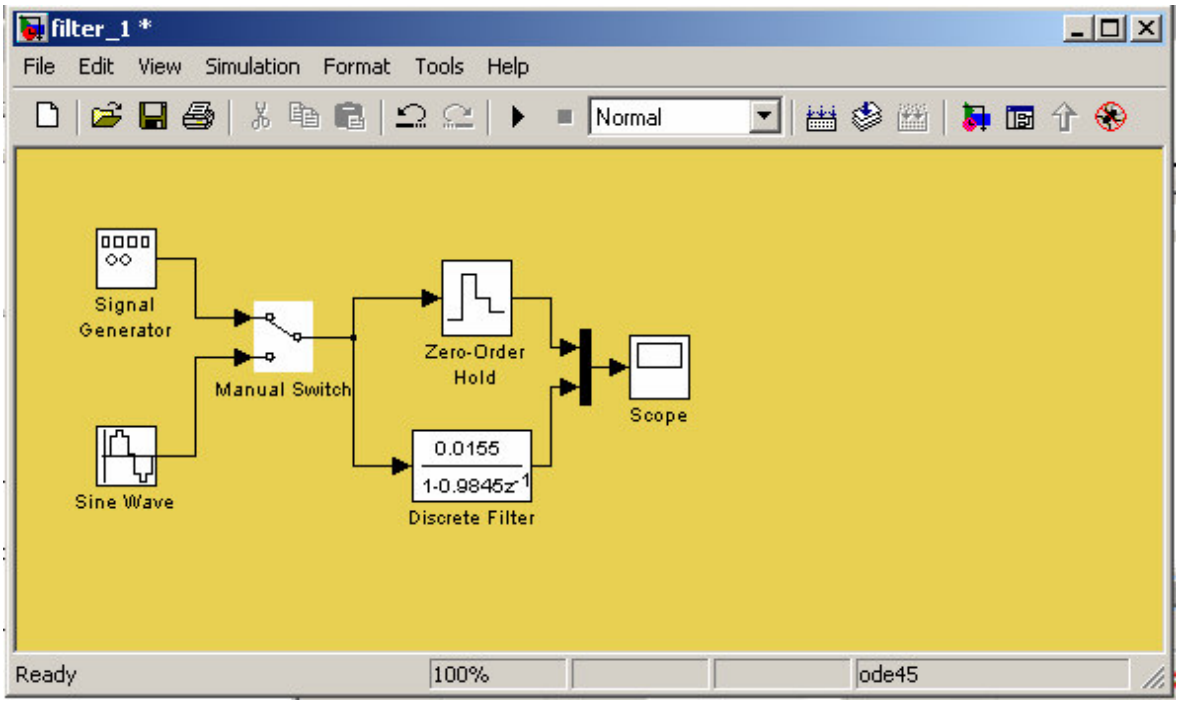


Figure 20: *The fully simulation of the DSPs is created in the Matlab Simulink. The scheme contains digital filter, signal generator, zero-holder and scope. The desired shape, amplitude and frequency of the input signal can be created in the signal generator. The desired digital filter can be entered into discrete filter. The output data can be seen in the scope.*

4.1.3.2 Result and discussion

*The frequency of the tested signal is **40 Hz**.*

The 40 Hz signal is passed with the minimal loss of the amplitude. The amplitude decline is caused by the quantization and rounding error. The phase of the signal is not changed. The filtered signal is compared with the original signal in the oscilloscope. The size of the

amplitude and phase of the filtered signal obtained by the oscilloscope is the same as in the filtered signal gained from the Matlab simulation (fig. 21 and 22).

The shape of the original square signal showed in the oscilloscope is not exactly square, because of the analogue low pass filter that is implemented before the DSC (fig. 21).

IIR 1-order LP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 40 Hz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + 0.0155 \cdot U \cdot (k \cdot T - 1)$

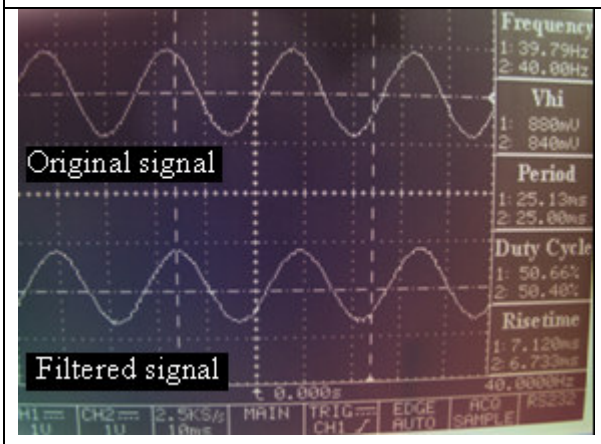
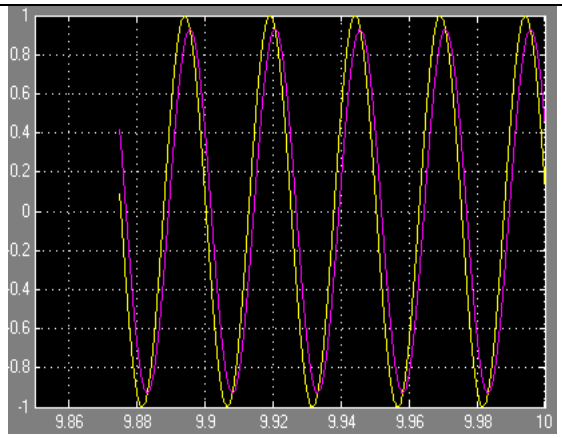
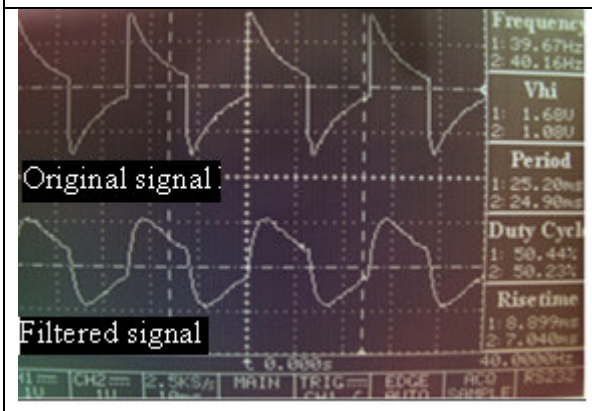
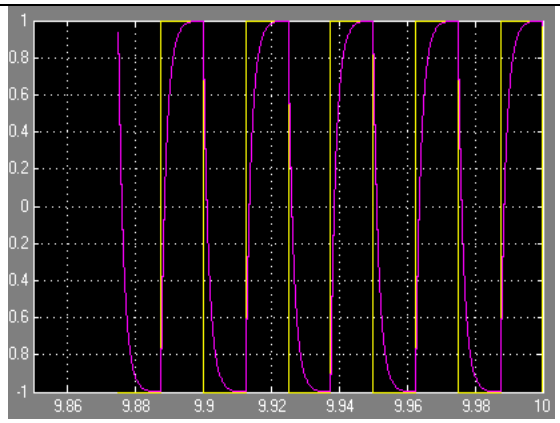
Oscilloscope	**Matlab Simulink**
	
	

Figure 21: *The IIR 1-order LP filter tested for frequency 40 Hz in the two different systems.*

IIR 1-order HP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 40 Hz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + U - U \cdot (k \cdot T - 1)$

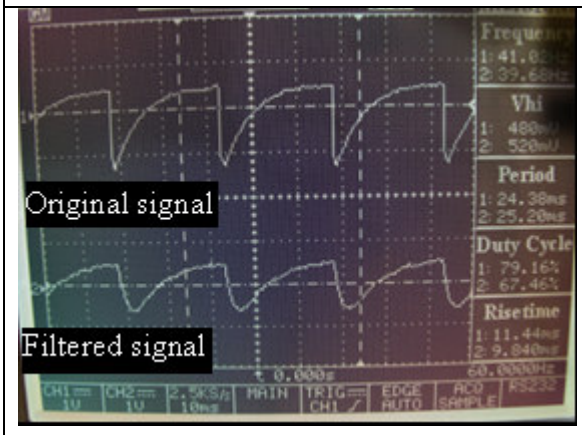
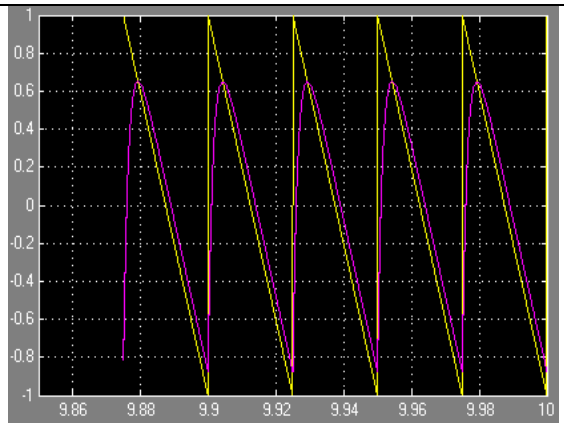
Oscilloscope	Matlab Simulink
	

Figure 22: *The IIR 1-order LP filter tested for the frequency 40 Hz in the two different systems.*

The frequency of the tested signal is ***1 kHz.***

IIR 1-order HP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 1 kHz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + U - U \cdot (k \cdot T - 1)$

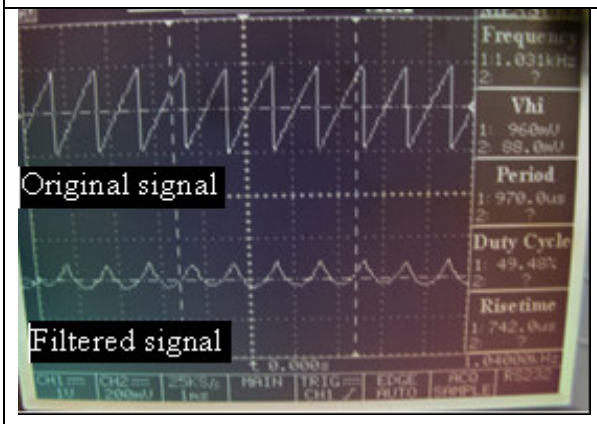
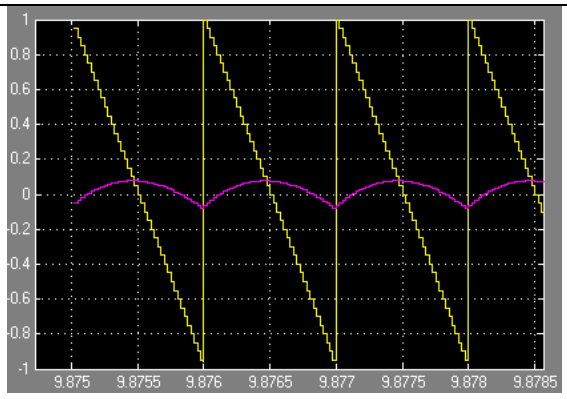
Oscilloscope	Matlab Simulink
	

Figure 23: *The IIR 1-order HP filter tested for frequency 1 kHz.*

For the input frequency 1 kHz, the signal is passed with huge loss of the amplitude. The amplitude of the filtered signal is approximately 10 smaller than amplitude of the original signal. The filtered signal is compared with the original signal in the oscilloscope. The phase of the signal is changed after the filtration is applied. The size of the amplitude and phase of the filtered signal obtained by the oscilloscope is the same as in the filtered signal gained from the Matlab simulation (fig. 23 and 24). The sawtooth generated by Matlab are flip side of the sawtooth generated by the generator.

IIR 1-order LP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 1 kHz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + 0.0155 \cdot U \cdot (k \cdot T - 1)$

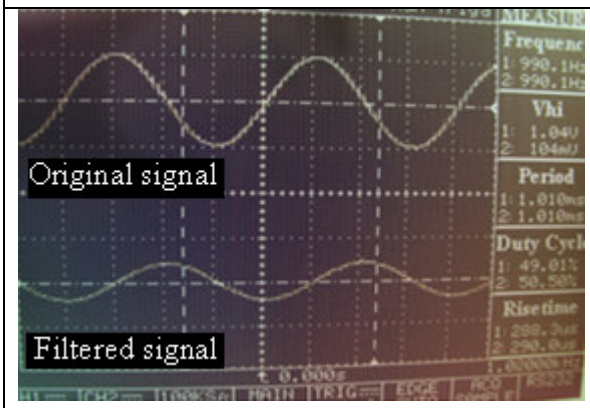
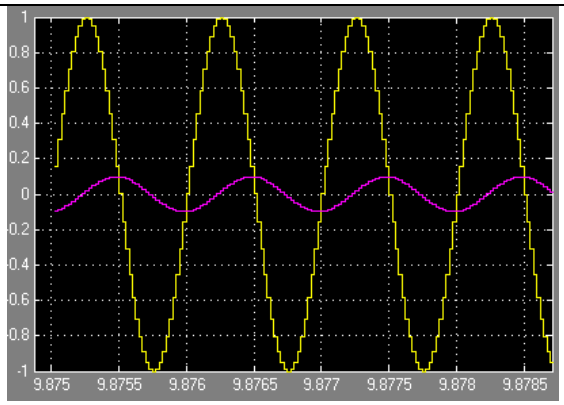
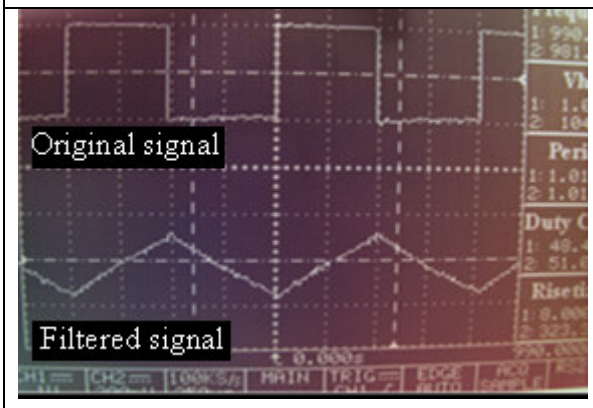
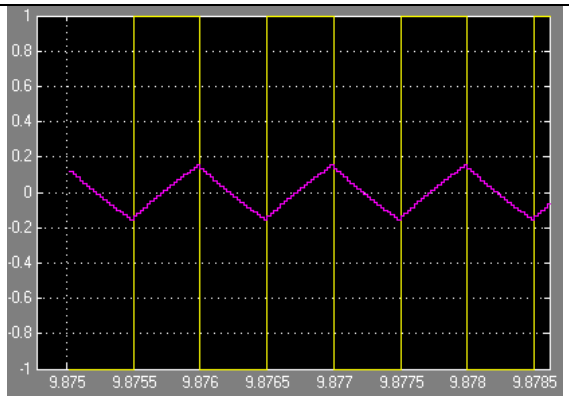
Oscilloscope	**Matlab Simulink**
	
	

Figure 24: *The IIR 1-order HP filter tested for frequency 1 kHz in the two different systems. The display resolution of the filtered signal is changed in the oscilloscope from 1V to 200 mV. It is owing to better visibility.*

4.2 IIR HP filter

4.2.1 Synthesis of the IIR 1-order HP filter

The first, second and third steps of the design are same as for the 1-order filter LP filter.

4. *Transfer function:*

The transfer function of the system is determined by the ratio of the voltage output and input in the equation 14 (the derivation of this equation is out of scope of this thesis). The transfer function of the low pass filter is defined in the equation 15.

$$\frac{V_{OUTPUT}}{V_{INPUT}} = \frac{R}{\frac{1}{s \cdot C} + R} = \frac{s \cdot C \cdot R}{1 + s \cdot C \cdot R} \tag{14}$$

$$G(s) = \frac{T \cdot s}{1 + T \cdot s} = \frac{\frac{T \cdot s}{T}}{\frac{1}{T} + \frac{T \cdot s}{T}} = \frac{s}{\frac{1}{T} + s} = \frac{Y(s)}{U(s)} \tag{15}$$

5. *The Z-transform:*

The Z-transform with the zero-order holder is used to gain the image of the transfer function in the complex plane (equation 16), with sampling frequency (f_s) at least two times higher as the highest used frequency (see sampling theorem). The image of the transfer function will contain the filter coefficients a_0 and b_0.

$$G(z,\varepsilon) = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{G(s)}{s} \right\} = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{s} \cdot \frac{s}{\frac{1}{T} + s} \right\} = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{\frac{1}{T} + s} \right\}$$

The Z - transform dictionary is used in the transform. Relative shift is chosen $\varepsilon = 0$.

$$G(z,\varepsilon)=\frac{z-1}{z}\cdot z_m\cdot\left\{\frac{1}{\frac{1}{T}+s}\right\}=\frac{z-1}{z}\cdot\left\{\frac{z\cdot e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\cdot\varepsilon\right)}}{z-e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}}\right\}=\frac{(z-1)\cdot e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\cdot\varepsilon\right)}}{z-e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}}= \tag{16}$$

$$=\frac{z-1}{z-e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}}=\frac{b_1\cdot z-b_0}{a_1\cdot z-a_0}$$

6. *The filter coefficients*:

The algorithm for the calculation of filter coefficients (equation 17) is derived from the equation 16 as:

$$b_0=1 \tag{17}$$

$$b_1=1$$

$$a_0=1-e^{\left(-\frac{1}{T}\cdot\frac{1}{f_s}\right)}$$

$$a_1=1$$

7. *HP IIR filter:*

The recursive equation for the 1-order HP IIR filter (equation 18) is gained by the z-transform (equation 16), where a_0 and b_0 are coefficients of the filter, u is input signal, y is output signal, and k are singular steps of the filtration.

Negative power of z is applied:

$$\frac{b_1\cdot z-b_0}{a_1\cdot z-a_0}=\frac{Y(z)}{U(z)}$$

$$\frac{b_1\cdot z\cdot z^{-1}-b_0\cdot z^{-1}}{a_1\cdot z\cdot z^{-1}-a_0\cdot z^{-1}}=\frac{Y(z)}{U(z)}$$

$$Y(kT) = a_0 \cdot Y \cdot (kT-1) + b_1 \cdot U - b_0 \cdot U \cdot (kT-1) \tag{18}$$

4.2.2 Calculation of the IIR 1-order HP filter

Task:

The HP filter can be used in the stereo equalizer when just high frequencies (trebles) are passed and low frequencies (basses) are banned. Everything that is lower than 100 Hz (f_c) will be attenuated. The HP filter can be designed as followed:

Solution:

The first, second and third steps of the design are same as for the 1-order LP filter.

4. *Transfer function:*

$$G(s) = \frac{0.0016 \cdot s}{1 + 0.0016 \cdot s} = \frac{Y(s)}{U(s)}$$

5. *The Z-transform:*

(ε = 0, zero-order holder, $f_s = 40kHz$)

$$G(z,\varepsilon) = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{0.0016 \cdot s}{s \cdot (1 + 0.0016 \cdot s)} \right\} = \frac{z-1}{z} \cdot \left\{ \frac{\frac{0.0016 \cdot s}{0.0016}}{s \cdot \left(\frac{1}{0.0016} + \frac{0.0016 \cdot s}{0.0016} \right)} \right\} =$$

$$= \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{1}{628 + s} \right\} = \frac{z-1}{z} \cdot z_m \cdot \left\{ \frac{z \cdot e^{\frac{-628}{40000_s} \cdot 0}}{z - \cdot e^{\frac{-628}{40000_s} \cdot 0}} \right\} = (z-1) \cdot \frac{1}{z - 0.9845} = \frac{z-1}{z - 0.9845}$$

4. *The filter coefficients:*

$b_0 = 1$

$b_1 = 1$

$$a_0 = -e^{\frac{-628}{40000_s}} = -0.9845$$

$$a_1 = 1$$

The control of the Z-transform calculation is also done in the Matlab by the command c2dm.

Whole command looks like:

num = [0.0016 0]
den = [0.0016 1]
[num1,den1] = C2DM(num,den,1/40000,'zoh')

The coefficients are also $a_0 = -0.9845$ and $b_0 = 1$.

5. *The HP LP IIR filter is (equation 19):*

Negative power of z is applied:

$$\frac{z-1}{z-0.9845} = \frac{Y(z)}{U(z)}$$

$$\frac{z \cdot z^{-1} - 1 \cdot z^{-1}}{z \cdot z^{-1} - 0.9845 \cdot z^{-1}} = \frac{Y(z)}{U(z)}$$

$$\frac{1 - z^{-1}}{1 - 0.9845 \cdot z^{-1}} = \frac{Y(z)}{U(z)}$$

$$Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + U - U \cdot (k \cdot T - 1) \qquad (19)$$

4.2.3 Implementation of the IIR 1-order HP filter

4.2.3.1 Theory

The designed and calculated IIR 1-order HP filter from the previous chapters is tested for the sine, square and sawtooth wave. First, the filter is tested for the input signal with 40 Hz

frequency. This signal shall be attenuated and the phase of the filtered signal shall be changed. Second, the 1 k Hz signal is send through the filter. This signal shall be passed with the minimal amplitude loss. The phase shall not be changed when the signal is passed through the filter. The recursion equation of the tested filter is in the equation 19. It is IIR 1-order HP filter designed for the $f_c = 100Hz$, and $f_s = 40kHz$.

4.2.3.2 Result and discussion

*The frequency of the tested signal is **40 Hz**.*

The 40 Hz original signal is passed with the expected loss of the amplitude. The filtered signal is compared with the original signal in the oscilloscope. The size of the amplitude and phase of the filtered signal obtained by the oscilloscope is almost the same as in the filtered signal gained from the Matlab simulation. As can be seen the signal phase is changed. The amplitude loss is approximately half. The shape of the original square signal showed in the oscilloscope is not exactly square, because of the analogue LP filter that is implemented before the DSC (fig. 25 and 26).

IIR 1-order HP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 40 kHz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + U - U \cdot (k \cdot T - 1)$

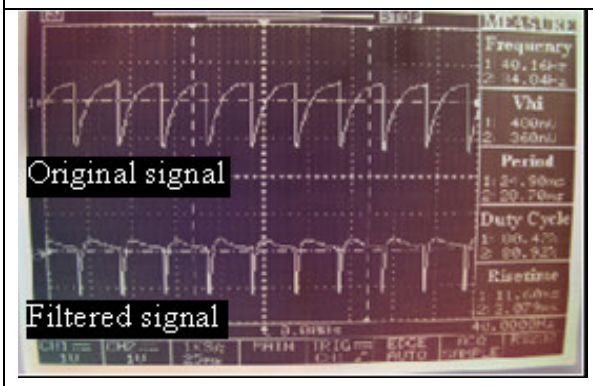
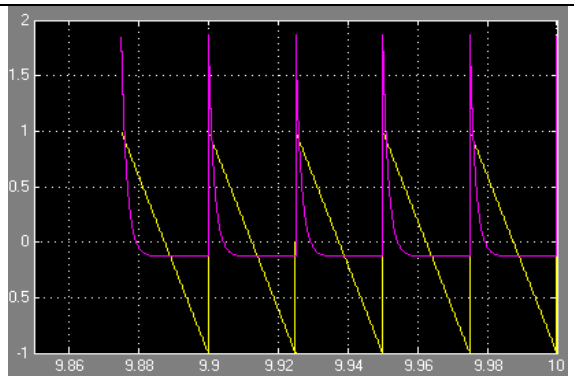
Oscilloscope	**Matlab Simulink**
	

Figure 25: *The IIR 1-order HP filter tested for the frequency 40 Hz. The Matlab sawtooth are flip side of the sawtooth in the oscilloscope.*

IIR 1-order HP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 40 Hz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + U - U \cdot (k \cdot T - 1)$

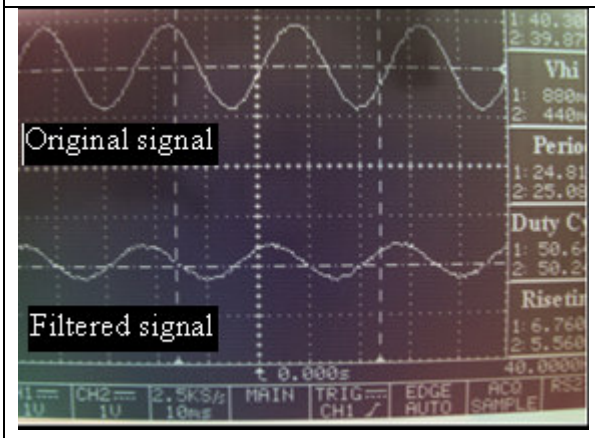
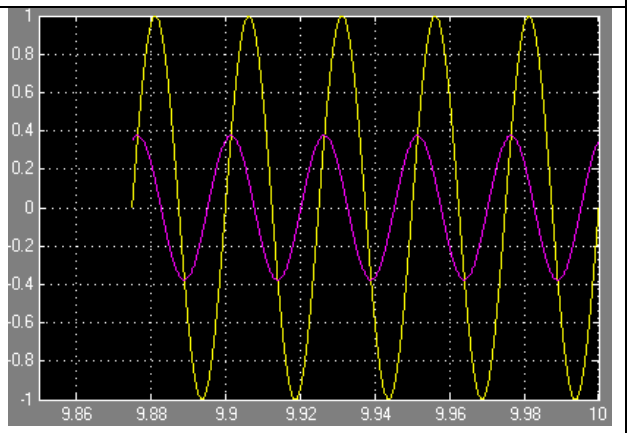
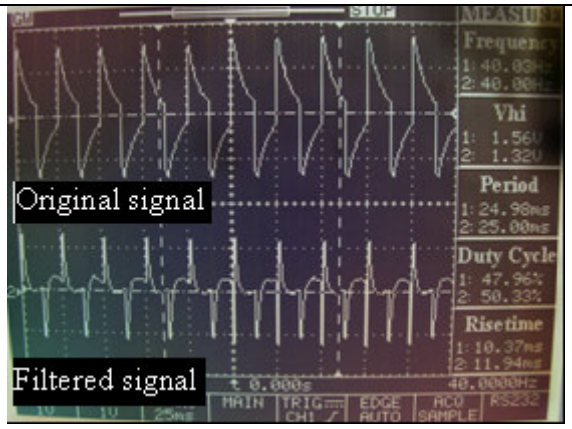
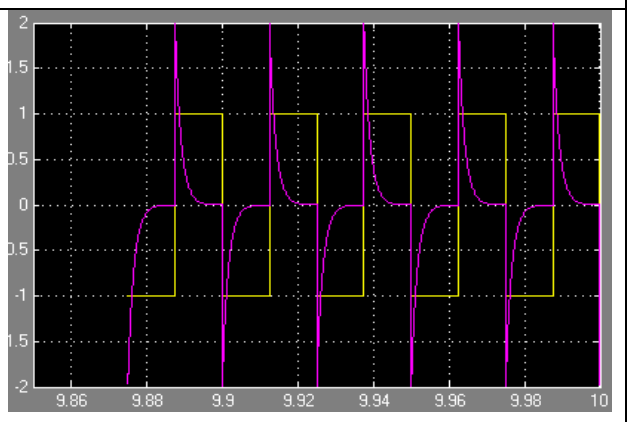
Oscilloscope	**Matlab Simulink**
	
	

Figure 26: *The IIR 1-order HP filter tested for the frequency 40 Hz in the two different systems.*

The frequency of the tested signal is ***1 kHz.***

The signal is passed for the input frequency 1 kHz with the minimal loss of the amplitude. The filtered signal is compared with the original signal in the oscilloscope. The size of the amplitude and the phase of the filtered signal obtained by the oscilloscope are same with the data obtained by the Matlab simulation (fig. 27).

IIR 1-order HP filter ($f_s = 40kHz$, $f_c = 100Hz$). The input signal freq. = 1 kHz.

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + U - U \cdot (k \cdot T - 1)$

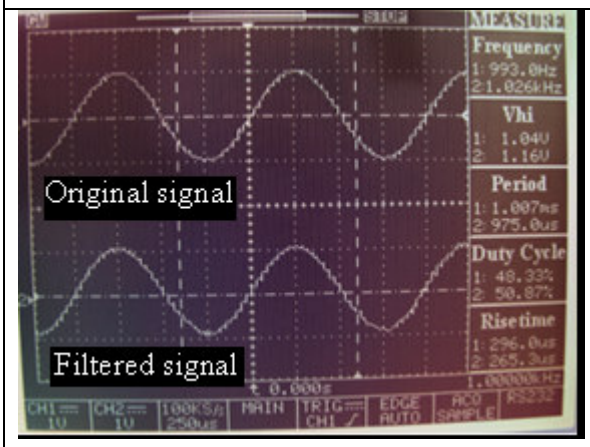
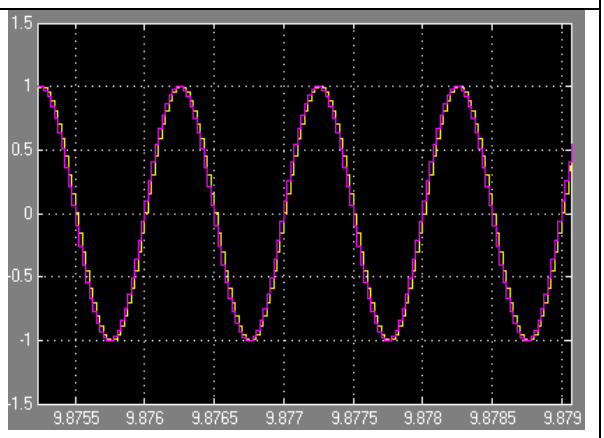
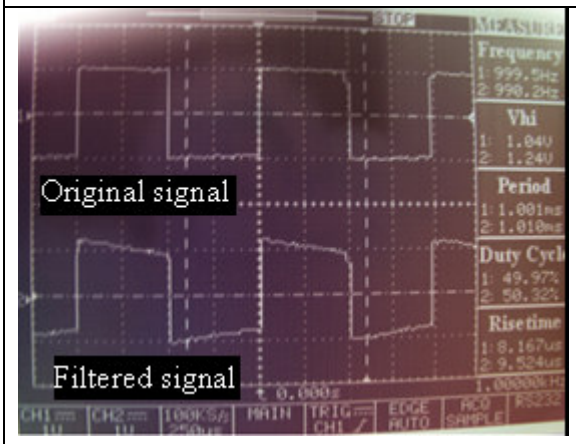
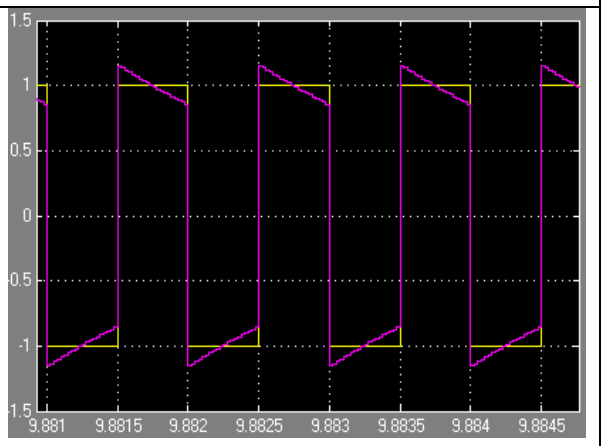
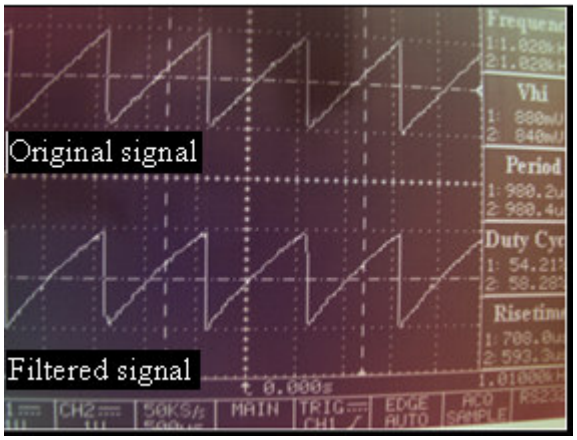
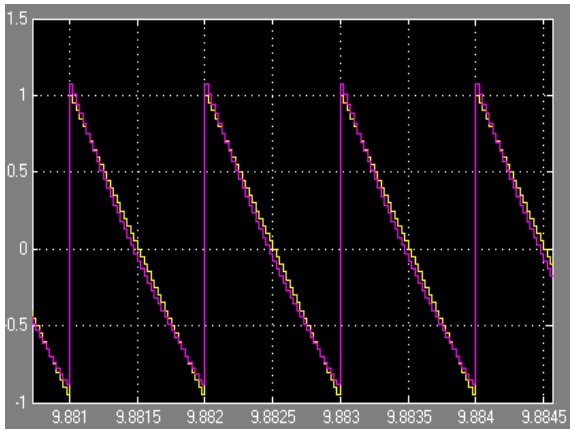
Oscilloscope	Matlab Simulink
	
	
	

Figure 27: *The IIR 1-order HP filter tested for the frequency 1 kHz in the two different systems. The phase of the filtered signal is not changed.*

4.3 Implementation of the IIR filter with the higher order

4.3.1 Theory

The higher order of the filter is made by cascading filters in series one, by one. The higher order of the filter has the better decline of the filtered signal. On the other hand, the rounding error is causing some loss of the information. This loss can be abandoned in the most application. But it is better to be aware about it during implementation of the filter with the higher order and try to find some appropriate combination between chosen order of the filter and rounding error. The decline of the IIR 6-order HP filter is tested for the 1 kHz input signal.

4.3.2 Result and discussion

The 1-order and 6-order LP filters are compared in the fig. 28. The 6-order filter has much better attenuation then 1-order filter. The 6-order filter has such a high decline, that every frequency higher then 37 Hz is totally attenuated. In other words the cut off frequency is moved from 100 Hz to 37 Hz. It is caused by the rounding error that occurs during filtration process. The rounding error of the 6-order filter is much higher then 1-order filter, owing to the more mathematical operations.

IIR 6-order LP filter ($f_s = 40kHz$, $f_c = 100Hz$ **). The input signal freq. = 40 Hz and 1kHz**

FILTER: $Y(kT) = 0.9845 \cdot Y \cdot (k \cdot T - 1) + 0.0155 \cdot U \cdot (k \cdot T - 1)$

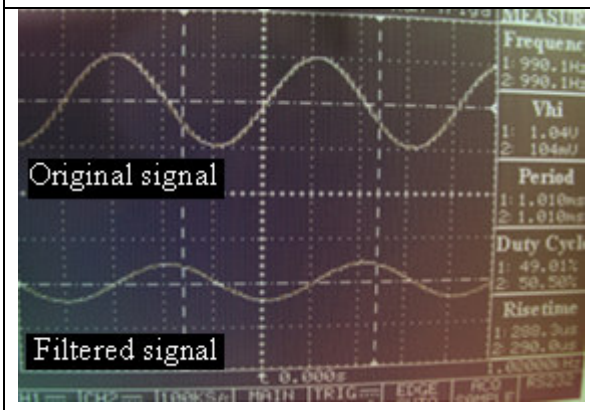
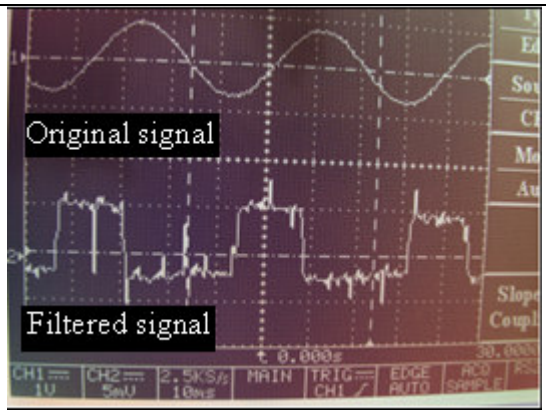
1-order	**6-order**
	

Figure 28: *The filtration of the1 kHz sine wave is done by the IIR 1-order and 6-order LP filter. The display resolution of the filtered signal is changed in the oscilloscope from 1V to 200 mV for the 1-order filter and to 5 mV for the 6-order filter. It is owing to better visibility.*

5 FILTER LIBRARIES

The DSP.h header file with universal libraries for the signal filtration was created. It can be used in any future application that will require the digital signal filtration. It contains these functions:

- The IIR LP and HP filters with the order from 1 to 6:

IIR_LP_HP_filter(filter,order,precision,offset,ADC_output,coeff,buff_0,buff_1,buff_2,buff_3,buff_4,buff_5)

The input parameters of the functions are:

filter = the type of the filter can be chosen (enter 0 for the LP and 1 for the HP filter),

order = the desired order of the filter can be chosen from 1 to 6,

offset = the desired offset can be set,

precision = the higher precision of the filtering can be achieved by the entering higher order of the decimal base (the lowest recommended order is 10000). The DSC 56F805 copes very well with the integer numbers, so it is better to convert the real numbers to the integer numbers. For example, filter coefficient are $a_0 = -0.9845$ and $b_0 = 0.0155$ and after conversion they will become:

$a_0 = -0.9845 \cdot 10000 = -9845$ and $b_0 = 0.0155 \cdot 10000 = 155$.

ADC_output = the input signal to the filter (for example from the ADC),

coeff = the filter coefficients calculated by the filter synthesis,

buff_0 - buff_5 = the buffers for the filter calculation process.

The return value of the functions is:

return(pom) = the filtered signal from the filter.

- The synthesis of the IIR LP and HP filter with the order from 1 to 6:

synthesis_IIR_1_LP_HP(filter, precision, coeff, cutoff_frequency, s_frequency)

The input parameters of the functions are:

filter, precision = are same as for the function IIR_LP_HP_filter,

coeff = the buffer where the filter coefficients calculated by this function are saved,

cutoff_frequency = the desired cut off frequency can be chosen,

s_frequency = the desired sampling frequency can be chosen.

The return values of the functions are:

coeff[0] = the calculated coefficient a_0,

coeff[1] = the calculated coefficient b_0.

- The calculation of the comparison value for the timer:

timer_period(40000)

The input parameter of the functions is:

sampling_frequency = the desired sampling frequency can be chosen,

The return value of the functions is:

cycle_repeating = the comparison value for the timer.

6 THE IMPLEMENTATION OF THE DSC 56F805

6.1 Flow chart of the exemplary program

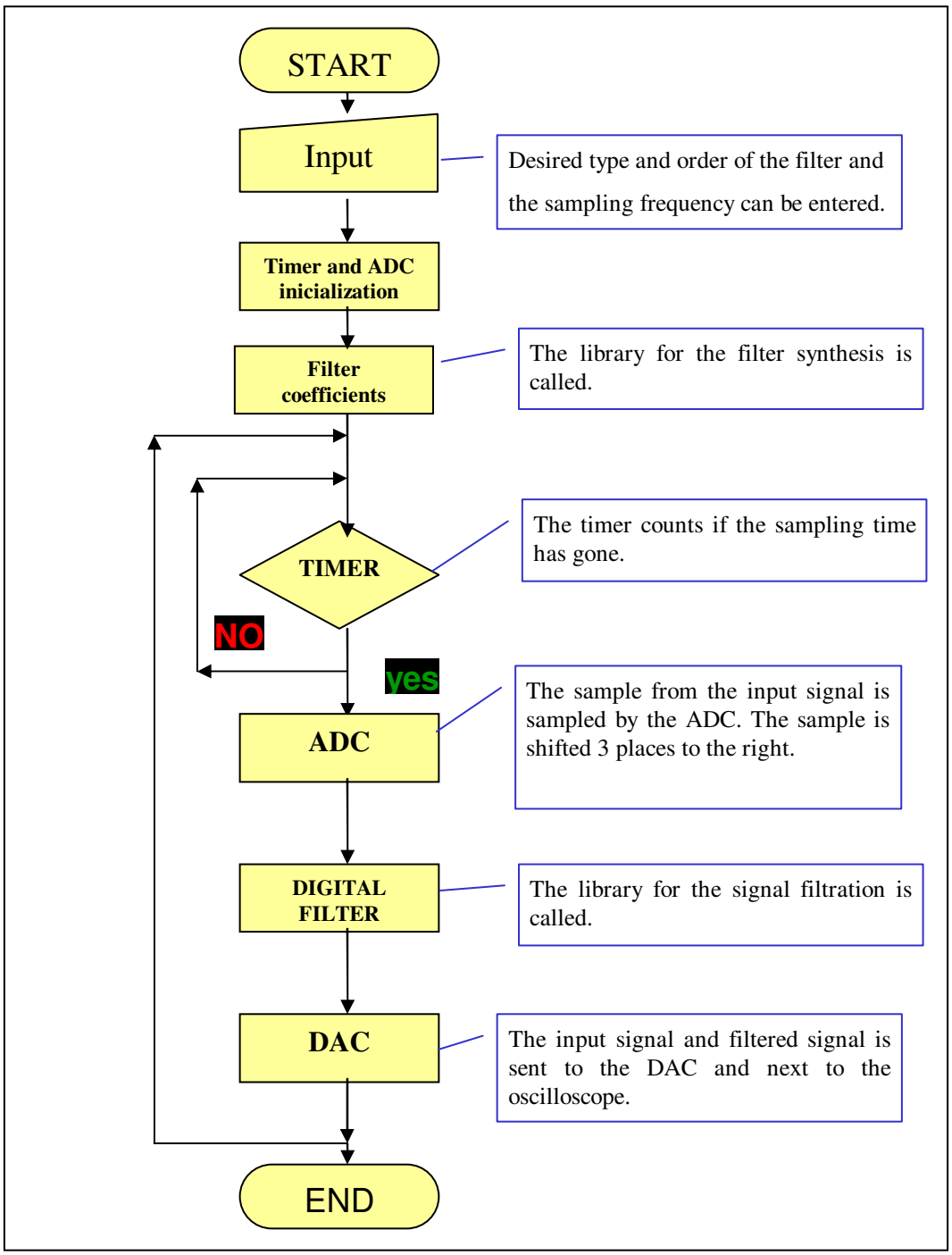


Figure 29: *The flow chart of the exemplary program.*

The flow chart is introduced in the figure 29. The filter libraries are called from the program. First, the synthesis of the filter and second the filtration. The program is more described in the header file that can be found in the appendix 1.

6.2 Implementation of the ADC

To implement the ADC is needful to set desired registers by the bit masks. The ADC registers that are used:

- ADC Control register 1,
- ADC Control Register,
- ADC Status Register,
- ADC Result Registers.

To initialize ADC is necessary to initialize the ADC Control register 1 and ADC Control Register (fig. 30). The operation of the ADC is stopped after the reset. So it is essential to set the STOP bit to logic 0. All other bits are left in the state as are after the reset (tab.3).

The START bit must be started in the every new cycle by setting logic 1.

ADC Control	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register 1 (ADCR1)	Read	0	STOP	0	SYNC	EOSIE	ZCIE	LLMTIE	HLMTIE	CHNCFG[3:0]				0	SMODE[2:0]		
X:ADC_BASE+$0	Write			START													
	Reset	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1

Figure 30: *The ADC Control register 1 is used for the ADC initialization.*

The clock divisor of the ADC is set by the combination of the zero, first, second and the third bits in the ADC Control Register. These bits can be set from 0 to 15 in the decimal mode by the DIV value. The equations and the single calculations of the DIV value are showed below in the (fig. 31).

DIV[3:0]	Clock Divisor Select
0–15	Clock Divisor Select Value = N = DIV + 1 F_{ADC} = (F_{IPR}) / 2N F_{ADC} = Analog-to-Digital Converter Frequency F_{IPR} = Interface Peripheral Bus Clock Frequency

$$F_{ADC} = 5\ MHz$$

$$F_{IRP} = 40\ MHz$$

$$5 = \frac{40}{2 \cdot N} \Rightarrow N = 4$$

$$4 = DIV + 1 \Rightarrow DIV = 3$$

ADC Control Register (ADCR2) X:ADC_BASE+$1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DIV[3:0]			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Figure 31: *The ADC Control register with the way how to set the clock of the timer.*

The ADC Status Register is used to know if the ADC conversion of the actual sample is finished (fig. 32). In this register is used zero bit that indicates if the channel is ready to be read (logic 1).

ADC Status Register (ADSTAT) X:ADC_BASE+$6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	CIP	0	0	0	EOSI	ZCI	LLMTI	HLMTI	RDY7	RDY6	RDY5	RDY4	RDY3	RDY2	RDY1	RDY0
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 32: *The Status Register is used to signalize the end of the ADC conversion.*

From the eight ADC Result Registers is enough to use just one (tab. 3). This register holds the sample gained from the ADC conversion. The size of the register is 12 bits. That makes 2^{12} = 4096 bit resolution of the ADC. The ADC Status and Result Registers are checked in the every cycle of the code (33).

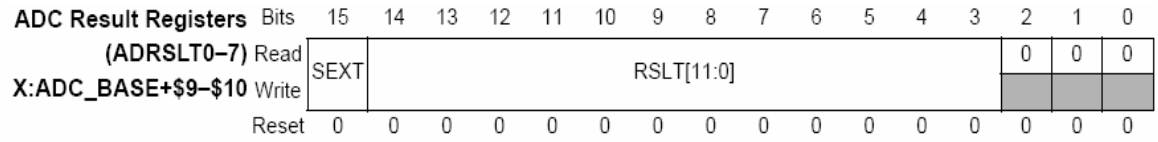


Figure 33: *The ADC Status Register is used to read sampled data from the ADC conversion.*

STOP bit ADCR1 (bit position 14)																	
Binary format																**Name**	**Operation**	**Hex format**
0	**1**	0	1	0	0	0	0	0	0	0	0	0	1	0	1	**ADCR1**		5005
1	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**Mask 1**	**AND**	bfff
0	**0**	0	1	0	0	0	0	0	0	0	0	0	1	0	1	**ADCR1**		1005
START bit ADCR1 (bit position 13)																		
0	1	**0**	1	0	0	0	0	0	0	0	0	0	1	0	1	**ADCR1**		5005
0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	**Mask 2**	**OR**	2000
0	0	**1**	1	0	0	0	0	0	0	0	0	0	1	0	1	**ADCR1**		3005
RDY0 bit ADSTAT (bit position 1)																		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	**ADSTAT**		0000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**Mask 3**	**AND**	0001
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	**ADSTAT**		0000

Table 3: *The table shows masks used for the correct ADC operation.*

6.3 Implementation of the timer

The timer A with channel 0 control is chosen. To implement the desired function of the timer is essential to set preferred registers by the bit masks.

The used timer registers are:

- TMR Control Register,
- TMR Status and Control Register,

- TMR Compare Register.

The TMR Control Register is used to initialize the timer before the main cycle in the code. The initialization is done by the hex mask 3022 (tab. 4). The count mode is set to count up the rising edges of primary source. The primary count source is chosen IP Bus clock divided by the one. The secondary count source is not used. Function ONCE is set to count repeatedly. The LENGTH is set to count until counter reaches the value saved in the CMP1 register (CMP1 register is used when counting up, for the counting down CMP2 register can be used). Count direction is set to counting up. Output mode is selected to set OFLAG output on the successful compare. The other bits are left in the reset mode (fig. 34).

TMR Control	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register (CTRL)	Read	Count Mode			Primary Count Source				Secondary Source		ONCE	LENGTH	DIR	Co Init	Output Mode		
TMRxn_BASE+	Write																
$6, $E, $16, or $1E	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 34: *The TMR Control Register is used to initialize the timer.*

The TMR Status and Control Register are applied to set the flag, when the successful compare has occurred (when the counting is finished). This register is tested in the every new cycle of the code by the hex mask 8000 (fig. 35).

TMR Status and	Bits	15	14	13	12	11	10	9	8	[7:6]	5	4	3	2	1	0
Control Register	Read	TCF	TCFIE	TOF	TOFIE	IEF	IEFIE	IPS	INPUT	Capture Mode	MSTR	EEOF	VAL		OPS	OEN
(SCR)	Write													FORCE		
TMRxn_BASE+	Reset	0	0	0	0	0	0	0	0	0 0	0	0	0	0	0	0

Figure 35: *The TMR Status and Control Register is used to signalize the end of the timer counting.*

The TMR Compare Register #1 is used for the storing of the comparison value that is compared with the value counted by the counter.

TMR Compare Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register #1 (CMP1) Read TMRxn_BASE+$0, Write	Comparison Value [15:0]															
$8, $10, $18 Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 36: *The TMR Compare Register stores the comparison value.*

CTRL																		
Binary format																**Name**	**Operation**	**Hex format**
0	0	**0**	**0**	0	0	0	0	0	0	**0**	0	0	0	**0**	0	**CTRL**		0000
0	0	**1**	**1**	0	0	0	0	0	0	**1**	0	0	0	**1**	0	**Mask 1**	**OR**	3022
0	0	**1**	**1**	0	0	0	0	0	0	**1**	0	0	0	**1**	0	**CTRL**		3022
TCF bit SCR (bit position 15)																		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**SCR**		0000
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**Mask 2**	**AND**	8000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**SCR**		0000

Table 4: *The table shows masks used for the desired timer operation.*

CONCLUSION

The principle aim of the thesis was the implementation of the basic digital filters in the DSC Motorola 56F805.

The low and high pass digital filters were chosen because they are the primary filters in the digital signal processing. The filter synthesis was successfully based on the 1-order IIR filters that simulate the characteristic of the analogue RC filters. The analogue RC filters were converted to the digital IIR filters by applying the Z-transform. The filters with the higher order were created by connecting more IIR 1-order filters in series.

The header file of the filter synthesis and IIR filters was coded using the C language. The DSC 56F805 implementation of the IIR filters have been tested on the real digital signal process composed from the waveform generator, DAC, DSP56F805 board and oscilloscope.

The Matlab Simulink was also used to create the digital filters. The data obtained from the Matlab were compared with the data gained by the experimental system. The comparison of the two different implementations of the same IIR filters has proven that filters operates correctly as expected through the theoretical knowledge. The header file of the filter synthesis and IIR filters can be used in any future applications which will require the filtration and synthesis of the signal.

The thesis also contains the background study of the digital signal processing and the method of implementation of the ADC and timer of the DSC 56F805. The header file and exemplary program for DSPs are included in the appendix 1 (CD-ROM).

The future challenge can be the design of the more complicated filters as the Butterworth, Bessel and Chebyshev filters.

TABLE OF USED LITERATURE

[1] SMÉKAL Z. a VÍCH R. (1998): Zpracování signálů pomocí signalových procesorů. RADIX. ISBN 80-7318-355-2

[2] SMITH S. (2007): The Scientist and Engineer's Guide to Digital Signal Processing [online]. Available from the webpage:

<http://www.dspguide.com/>

[3] IFEACHOR E.C. and JERVIS B.W. (1993): Digital Signal Processing / A Practical Approach. Addison-Wesley Publisher Ltd. ISBN 0-201-54413-X

[4] BORES CH. (2007): Introduction to DSP [online]. Available from the webpage:

<http://www.bores.com/courses/intro/index.htm>

[5] Quantization error [online]. Wikimedia Foundation. Available from the webpage:

<http://en.wikipedia.org/wiki/Quantization_noise>

[6] Digital signal processor [online]. Wikimedia Foundation. Available from the webpage:

<http://en.wikipedia.org/wiki/Digital_signal_processor>

[7] Transfer function [online]. Wikimedia Foundation. Available from the webpage:

<http://en.wikipedia.org/wiki/Transfer_function>

[8] Bode plot [online]. Wikimedia Foundation. Available from the webpage:
<http://en.wikipedia.org/wiki/Bode_plot>

[9] GDS820S oscilloscope manual [online]. Available from the webpage:
<http://www.testequipmentdepot.com/instek/oscilloscopes/gds820_840series.htm>

[10] M631 Arbitrary Waveform manual [online]. Available from the webpage:
<Generatorhttp://www.etcsk.com/products/m631/index.php?lang=eng>

[11] DSC Motorola 56F805 [online]. Available from the webpage:
<http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=DSP56F805&nodeId=01279562926045>

[12] Freescale [online]. Available from the webpage:
<http://www.freescale.com>

[13] Filter design [online]. Wikimedia Foundation. Available from the webpage:
<http://en.wikipedia.org/wiki/Filter_design>

[14] SMÉKAL Z. a VÍCH R. (2000): Číslicové Filtry. Academia. ISBN 80-200-0761-X

[15] Waveform [online]. Wikimedia Foundation. Available from the webpage:
<http://en.wikipedia.org/wiki/Waveform>

[16] RC circuit [online]. Wikimedia Foundation. Available from the webpage:
<http://en.wikipedia.org/wiki/RC_circuit>

[17] Frequency Response and Active Filters [online]. Available from the webpage:
<http://www.swarthmore.edu/NatSci/echeeve1/Ref/FilterBkgrnd/Filters.html>

TABLE OF USED SYMBOLS AND ABBREVIATIONS

f_s	Sampling Frequency
f_c	Cut Off Frequency
f_{MAX}	The highest used frequency
ADC	Analogue to Digital Convector.
CMOS	Complementary Metal–Oxide–Semiconductor
DAC	Digital to Analogue Convector
DSP	Digital Signal Processor.
DSPs	Digital Signal Processing
FIR	Finite Impulse Response
FIR	Infinite Impulse Response
HP	High pass
JTAG	Joint Test Action Group
LP	Low pass
LPT	Line Print Terminal
LTI	Linear-Time-Invariant-System
NMOS	N-type Metal-Oxide-Semiconductor
PC	Personal Computer
RAM	Random Access Memory
RC	Resistor Capacitor circuit.
SPI	Serial Peripheral Interface Bus
T	Time constant

TABLE OF THE FIGURES

***Figure 1**: The diagram of general real-time DSP system* 13
***Figure 2:** The sampling of the analogue signal* 14
***Figure 3:** The aliasing effect* 15
***Figure 4:** Quantization erro* 16
***Figure 5:** The Bode diagram.* 20
***Figure 6:** The basic parameters of the oscilloscope.* 21
***Figure 7:** The basic parameters of the waveform generator.* 21
***Figure 8:** The system map of the experimental DSPs system.* 22
***Figure 9:**The generator software* 23
***Figure 10:** The Block Diagram of the DSC 56F805* 25
***Figure 11:** The LP RC filter.* 27
***Figure 12:** The RC filter in the time domain* 27
***Figure 13:** The Bode diagram for the LP RC filter* 28
***Figure 14:** The HP RC filter.* 29
***Figure 15:** The Bode diagram for the LP RC filter* 30
***Figure 16:** IIR1-order LP recursive and RC filter* 32
***Figure 17:** The 4-order analogue LP filter.* 33
***Figure 18:** The 2-order IIR LP filter* 33
***Figure 19:** The different shapes of the waveforms* 41
***Figure 20:** The fully simulation of the DSPs is created in the Matlab Simulink.* 42
***Figure 21:** The IIR 1-order LP filter tested for frequency 40 Hz* 43
***Figure 22:** The IIR 1-order LP filter tested for the frequency 40 Hz.* 44
***Figure 23:** The IIR 1-order HP filter tested for frequency 1 kHz* 44
***Figure 24:** The IIR 1-order HP filter tested for frequency 1 kHz* 45
Figure 25: The IIR 1-order HP filter tested for the frequency 40 Hz. 51
***Figure 26:** The IIR 1-order HP filter tested for the frequency 40 Hz* 51
***Figure 27:** The IIR 1-order HP filter tested for the frequency 1 kHz s.* 53
***Figure 28:** The IIR 1-order and 6-order LP filter.* 54
***Figure 29:** The flow chart of the exemplary program.* 58

Figure 30: *The ADC Control register 1* 59
Figure 31: *The ADC Control register* 60
Figure 32: *The Status Register is used to signalize the end of the ADC conversion.* 60
Figure 33: *The ADC Status Register used to read sampled.* 61
Figure 34: *The TMR Control Register used to initialize the timer.* 62
Figure 35: *The TMR Status and Control Register* 62
Figure 36: *The TMR Compare Register stores the comparison value.* 63

LIST OF TABLES

Table 1: *The bitwise operations.....19*

Table 2: *The FIR and IIR filters.....31*

Table 3: *The bitwise masks used to set Timer.....61*

Table 4: *The bitwise operations used to set timer.....63*

TABLE OF APPENDIXES

APPENDIX A I: CD-ROM (CONTAINS THE HEADER FILE DSP.H WITH THE FILTER LIBRARIES AND EXEMPLARY PROGRAM)